
PgnHelper

Release 0.10.0

Ferdinand Mosca

Jul 16, 2022

CONTENTS:

1	Features	3
1.1	1. Add ECO codes, Opening and Variation names	3
1.2	2. Sort games	4
1.3	3. Generate round-robin table	4
1.4	4. Calculate the Elo rating change of a player	5
1.5	5. Generate swiss table	6
1.6	6. Generate opening stats	7
	Python Module Index	41
	Index	43

PgnHelper is an application that can sort games; add ECO codes, opening and variation names; generates a round-robin, swiss and standing tables with rating change and tie-break scores; and opening stats.

First install the pgnhelper package see the *Installation* guide.

FEATURES

1.1 1. Add ECO codes, Opening and Variation names

Command line:

```
pgnhelper addeco --inpgnfn manila1974.pgn --outpgnfn out_manila1974.pgn --inecopgnfn eco.  
↪pgn
```

Sample output:

```
[Event "Manila"]  
[Site "Manila"]  
[Date "1974.??.??"]  
[Round "10"]  
[White "Naranja, Renato"]  
[Black "Petrosian, Tigran V"]  
[Result "1/2-1/2"]  
[BlackElo "2640"]  
[ECO "A15"]  
[ECOT "D90"]  
[Opening "English"]  
[OpeningT "Gruenfeld"]  
[VariationT "Three knights variation"]  
[WhiteElo "2395"]  
  
1. c4 Nf6 2. Nf3 g6 3. d4 Bg7 4. Nc3 d5 5. cxd5 Nxd5 6. Bd2 Nb6 7. Qc2 Nc6  
8. Rd1 O-O 9. e3 Bf5 10. Qc1 a5 11. Be2 a4 12. O-O Qc8 13. d5 Nb8 14. e4 Bg4  
15. Bh6 c6 16. Bxg7 Kxg7 17. Qe3 N8d7 1/2-1/2
```

The new ECOT, OpeningT and VariationT (T=Transposition) are based from the input eco.pgn file. See the *Usage* section for further informations.

The ECO tag is based from the first 2 moves of the game while the ECOT is based from the first 12 moves of the game.

1.2 2. Sort games

Listing 1: Sort games by eco tag.

```
pgnhelper sort --inpgnfn airthings.pgn --outpgnfn sorted_airthings.pgn --sort-tag eco --
↳sort-direction hightolow
```

Listing 2: Sort games by white tag.

```
pgnhelper sort --inpgnfn airthings.pgn --outpgnfn sorted_airthings.pgn --sort-tag white -
↳-sort-direction hightolow
```

Available sort tags:

```
event, site , date, round, white, black, eco, ecot, plycount
```

1.3 3. Generate round-robin table

Listing 3: Round-robin with normal scoring

```
pgnhelper roundrobin --inpgnfn superbet_classic_2022_bucharest.pgn --output scb2022.txt
```

Sample output with rating change and tie-break scores DE, Wins and SB:

Rank	Name	Rating	RChg	1	2	3	4	5	6	7	8	↳
↳9	10 Games	Score	Score%	DE	Wins	SB						
1	Aronian, Levon	2765	9.50	x	0.5	1.0	1.0	0.5	0.5	0.5	0.5	0.
↳5	0.5	9	5.5	61.11	1.5	2	24.75					
2	So, Wesley	2776	7.93	0.5	x	0.5	0.5	0.5	0.5	1.0	0.5	1.
↳0	0.5	9	5.5	61.11	1.0	2	23.50					
3	Vachier-Lagrave, Maxime	2750	11.64	0.0	0.5	x	0.5	1.0	0.5	0.5	1.0	0.
↳5	1.0	9	5.5	61.11	0.5	3	23.00					
4	Dominguez Perez, Leinier	2753	1.21	0.0	0.5	0.5	x	0.5	1.0	0.5	0.0	1.
↳0	0.5	9	4.5	50.00	1.5	2	19.50					
5	Caruana, Fabiano	2786	-3.49	0.5	0.5	0.0	0.5	x	0.5	0.5	0.5	1.
↳0	0.5	9	4.5	50.00	1.0	1	19.25					
6	Deac, Bogdan-Daniel	2671	12.62	0.5	0.5	0.5	0.0	0.5	x	0.5	0.5	0.
↳5	1.0	9	4.5	50.00	0.5	1	19.75					
7	Nepomniachtchi, Ian	2773	-6.64	0.5	0.0	0.5	0.5	0.5	0.5	x	1.0	0.
↳0	0.5	9	4.0	44.44	1.0	1	18.00					
8	Firouzja, Alireza	2804	-11.04	0.5	0.5	0.0	1.0	0.5	0.5	0.0	x	0.
↳5	0.5	9	4.0	44.44	0.0	1	18.00					
9	Mamedyarov, Shakhriyar	2759	-9.65	0.5	0.0	0.5	0.0	0.0	0.5	1.0	0.5	↳
↳x	0.5	9	3.5	38.89	0.5	1	15.50					
10	Rappport, Richard	2776	-12.07	0.5	0.5	0.0	0.5	0.5	0.0	0.5	0.5	0.
↳5	x	9	3.5	38.89	0.5	0	15.75					

Listing 4: Round-robin with armageddon tie-break

```
gnhelper roundrobin --inpgnfn norwaychess.pgn --armageddon-file norwaychess_arm.pgn --
↳output norwaychess.html --win-point 3.0 --win-point-arm 1.5 --loss-point-arm 1.0 --
↳show-max-score
```

(continues on next page)

(continued from previous page)

Sample output:

Rank	Name	Rating	RChg	1	2	3	4	5	6	7	8	9
↪ 10	Games	Score	MaxScore	Score%	DE	Wins						
1	Carlsen, Magnus	2864	-0.05	x	3.0	1.0	1.5	1.0	3.0	1.5	1.0	3.0
↪ 1.5	9	16.5	27.0	61.11	0.0	0						
2	Mamedyarov, Shakhriyar	2759	9.31	0.0	x	3.0	1.0	1.5	3.0	1.5	1.5	1.0
↪ 3.0	9	15.5	27.0	57.41	0.0	0						
3	Anand, Viswanathan	2751	5.44	1.5	0.0	x	3.0	1.0	1.5	3.0	1.5	1.5
↪ 1.5	9	14.5	27.0	53.70	0.0	0						
4	Vachier-Lagrave, Maxime	2750	5.58	1.0	1.5	0.0	x	3.0	1.5	1.5	3.0	1.0
↪ 1.5	9	14.0	27.0	51.85	0.0	0						
5	So, Wesley	2776	-3.09	1.5	1.0	1.5	0.0	x	1.5	1.0	1.5	3.0
↪ 1.5	9	12.5	27.0	46.30	0.0	0						
6	Giri, Anish	2761	-0.97	0.0	0.0	1.0	1.0	1.0	x	1.5	3.0	3.0
↪ 1.5	9	12.0	27.0	44.44	0.0	0						
7	Topalov, Veselin	2730	-1.60	1.0	1.0	0.0	1.0	1.5	1.0	x	1.5	1.5
↪ 1.0	9	9.5	27.0	35.19	1.5	0						
8	Tari, Aryan	2654	8.79	1.5	1.0	1.0	0.0	1.0	0.0	1.0	x	1.0
↪ 3.0	9	9.5	27.0	35.19	1.0	1						
9	Radjabov, Teimour	2753	-14.84	0.0	1.5	1.0	1.5	0.0	0.0	1.0	1.5	x
↪ 1.5	9	8.0	27.0	29.63	0.0	0						
10	Wang, Hao	2744	-8.57	1.0	0.0	1.0	1.0	1.0	1.0	1.5	0.0	1.0
↪ x	9	7.5	27.0	27.78	0.0	0						

1.4 4. Calculate the Elo rating change of a player

Game result:

```
[Event "FIDE Candidates 2022"]
[Site "Madrid ESP"]
[Date "2022.06.17"]
[Round "1"]
[White "Ding Liren"]
[Black "Nepomniachtchi,I"]
[Result "0-1"]
[WhiteElo "2806"]
[BlackElo "2766"]
[EventDate "2022.06.16"]
[ECO "A20"]
```

Listing 5: Calculate the rating change of Nepomniachtchi.

```
1 import pgnhelper.elo
2
3 white_rating = 2806
4 black_rating = 2766
5 white_point = 0
6 black_point = 1
```

(continues on next page)

(continued from previous page)

```

7 expected_score = pgnhelper.elo.expected_score(black_rating, white_rating)
8 k = 10
9 rating_change = k * (black_point - expected_score)
10 print(rating_change) # 5.573116337622928

```

Listing 6: Calculate the rating change of Ding Liren.

```

1 import pgnhelper.elo
2
3 white_rating = 2806
4 black_rating = 2766
5 white_point = 0
6 black_point = 1
7 expected_score = pgnhelper.elo.expected_score(white_rating, black_rating)
8 k = 10
9 rating_change = k * (white_point - expected_score)
10 print(rating_change) # -5.573116337622928

```

1.5 5. Generate swiss table

```

pgnhelper swiss --inpgnfn "./pgn/fide_grand_swiss_2021_riga.pgn" --output "fide_swiss.txt"
↪
# pgnhelper swiss --inpgnfn "./pgn/fide_grand_swiss_2021_riga.pgn" --output "fide_swiss.
↪html"
# pgnhelper swiss --inpgnfn "./pgn/fide_grand_swiss_2021_riga.pgn" --output "fide_swiss.
↪csv"

```

Rank	Name	Rating	RChg	R1	R2	R3	R4	R5	R6	
↪R7	1	Firouzja, Alireza	2770	11.31	93W1	35B1	7W1	4B=	6W=	8B=
↪33W1	2	Caruana, Fabiano	2800	1.06	55W1	18B=	36W=	35B=	9W1	33B=
↪13W=	3	Oparin, Grigoriy	2654	21.62	62W=	78B1	17W=	70B=	26W=	5B=
↪73W1	4	Yu, Yangyi	2704	9.20	5W=	63B1	61W1	1W=	49B=	13B=
↪15W=	5	Keymer, Vincent	2630	22.06	4B=	27W1	25B=	10W=	59B=	3W=
↪24B0	6	Vachier-Lagrave, Maxime	2763	2.12	52B=	51W1	71B=	60W1	1B=	32W1
↪31B=	7	Predke, Alexandr	2666	14.22	46B1	54W1	1B0	47W=	42B1	31W0
↪57B1	8	Shirov, Alexei	2659	16.99	43B=	87W1	50B=	92W1	36B1	1W=
↪14B=	9	Howell, David W L	2658	15.43	82B1	92W=	72B=	50W=	2B0	54W1
↪47B1	10	Sargissian, Gabriel	2664	10.56	72W=	83B=	94W1	5B=	70W1	15B=
↪18W=										

(continues on next page)

(continued from previous page)

11	Anton Guijarro, David	2658	13.25	37W=	70B=	62W1	17B0	98W1	91B1	↵			
↵19W=	21B1	6W=	7B=	13W=	11	7.0	63.64	61.0	65.0	39.25	0.0	4	2
12	Korobov, Anton	2690	6.07	48B=	20W1	47B=	42W=	50B1	49W=	↵			
↵17B=	41W1	9B0	34W=	31B1	11	7.0	63.64	60.5	66.0	41.50	0.0	4	2
13	Sevian, Samuel	2654	15.89	80B=	84W1	92B=	72W1	32B=	4W=	↵			
↵2B=	53W1	30B=	10W=	11B=	11	7.0	63.64	60.5	64.5	39.75	0.0	3	0
14	Esipenko, Andrey	2720	3.08	58B=	42W=	81B=	63W1	71B=	56W1	↵			
↵8W=	9B0	64B1	31W1	15B=	11	7.0	63.64	60.0	64.5	40.00	0.0	4	1
15	Deac, Bogdan-Daniel	2643	15.05	89B=	104W=	106B1	53W=	29B1	10W=	↵			
↵4B=	25W=	18B=	38B1	14W=	11	7.0	63.64	60.0	63.0	39.25	0.0	3	3
16	Artemiev, Vladislav	2699	4.40	76B1	50W=	75B=	31W=	51B=	47W=	↵			
↵48B=	52W=	49B1	17W=	34B1	11	7.0	63.64	56.5	61.5	39.00	0.0	3	3
...													

Tie-breaks:

```

TB1 = Buchholz Cut 1
TB2 = Buchholz
TB3 = Sonneborn-Berger
TB4 = Direct Encounter
TB5 = Number of wins
TB6 = Number of wins as black

```

1.6 6. Generate opening stats

```

>>> import pgnhelper.eco
>>> import pgnhelper.record
>>> df = pgnhelper.eco.get_opening_stats("./pgn/candidates_zurich_1953.pgn")
>>> df

```

	Opening	Count	Count%
0	King's Indian Defence	44	20.95
1	Nimzo-Indian	41	19.52
2	Sicilian	23	10.95
3	English	18	8.57
4	Queen's Gambit Declined	16	7.62
5	Queen's Indian Defence	12	5.71
6	Ruy Lopez	10	4.76
7	Old Indian	7	3.33
8	French	6	2.86
9	Catalan	4	1.90
10	King's Indian	4	1.90
11	QGD Slav	4	1.90
12	Benoni	3	1.43
13	Dutch	3	1.43
14	Gruenfeld	3	1.43
15	QGA	3	1.43
16	Zukertort	3	1.43

(continues on next page)

(continued from previous page)

17	Caro-Kann	2	0.95
18	Neo-Gruenfeld	2	0.95
19	Queen's pawn game	2	0.95

Note: Your game must have an opening info in the header.

1.6.1 Installation

Virtual environment

I would recommend this kind of installation as opposed to global because the clash of library versions installed are avoided. Also it is easy to cleanup the installation, you just have to delete the virtual environment folder.

Typical setup on windows 10.

1. Open command prompt or powershell:

```
PS C:\Users\ferdi>
```

2. cd to C drive or other drive if you want to install in other drive:

```
PS C:\Users\ferdi> cd c:\
```

3. Check if you have python:

```
PS C:\> python --version
```

Note: Be sure you use Python version 3.7, 3.8 ... or later.

4. Create a folder in this example I will name it mypgnhelper:

```
PS C:\> mkdir mypgnhelper
```

5. cd to this folder:

```
PS C:\> cd mypgnhelper
```

6. Create venv folder for virtual environment:

```
PS C:\mypgnhelper> python -m venv venv
```

Note: The folder venv is now under the mypgnhelper folder. There is python installed under venv. You may delete this folder if you no longer need it.

7. Activate the virtual environment:

```
PS C:\mypgnhelper> ./venv/scripts/activate
```

8. Update pip:

```
(venv) PS C:\mypgnhelper> python -m pip install -U pip
```

9. Install the pgnhelper:

```
(venv) PS C:\mypgnhelper> pip install pgnhelper
```

Note: All the dependent modules of pgnhelper are also installed. See the dependent libraries section below.

10. Check the installation by checking its version:

```
(venv) PS C:\mypgnhelper> pgnhelper -v
```

When you work with pgn file you need to activate the virtual environment as in step 7 if it is not activated yet like when you comeback after computer restart.

If you no longer need the pgnhelper, just delete the mypgnhelper folder.

Global

1. Open command prompt or powershell and run as administrator:

```
PS C:\WINDOWS\system32>
```

2. cd to c drive:

```
PS C:\WINDOWS\system32> cd c:\
```

3. Check if you have python:

```
PS C:\> python --version
```

4. Install the package:

```
PS C:\> pip install pgnhelper
```

5. Test it:

```
PS C:\> pgnhelper -h
```

Dependent libraries

pgnhelper is dependent on the following libraries:

```
python chess  
pandas  
pretty-html-table  
pytest
```

They are installed automatically when pgnhelper is installed.

1.6.2 Uninstallation

Uninstall the package with:

```
pip uninstall pgnhelper
```

The dependent libraries installed by pgnhelper are not uninstalled automatically. You have to uninstall it manually.

Uninstall other libraries:

```
pip uninstall chess
pip uninstall pandas
pip uninstall pretty-html-table
pip uninstall pytest
```

Note: Do not uninstall the other libraries if you still need it specially if you install pgnhelper globally or not using virtual environment.

1.6.3 Help

General help

Open your command prompt or powershell and type:

```
pgnhelper --help
```

```
usage: pgnhelper [-h] [-v] {sort,addeco,roundrobin,standing} ...

positional arguments:
  {sort,addeco,roundrobin,standing}
    sort                Sort the games from the given pgn file based on the given game.
    ↪tags. e.g. pgnhelper sort mygames.pgn --outpgnfn out.pgn --sort-tag opening --sort-
    direction hightolow
    addeco              Add eco and ecot codes, opening and variation names to the input.
    ↪pgn file. The eco, opening etc. are from the given input file eco.pgn. e.g.
    pgnhelper addeco --inpgnfn mygames.pgn --inecopgnfn eco.pgn --
    ↪outpgnfn out.pgn
    roundrobin         Generate round-robin table results from the input pgn file. The
    ↪output can be html, csv and txt. e.g. pgnhelper roundrobin --inpgnfn candidates.pgn
    --output candidates.html
    standing           Generates a standings from the input pgn file. The output can be
    ↪html, csv and txt. e.g. pgnhelper standing --inpgnfn candidates.pgn --output
    candidates.html

options:
  -h, --help          show this help message and exit
  -v, --version       show program's version number and exit
```

Sort help

```
pgnhelper sort --help
```

```
usage: pgnhelper sort [-h] --inpgnfn INPGNFN --outpgnfn OUTPGNFN [--sort-tag SORT_TAG] [-
↳--sort-direction SORT_DIRECTION] [--encoding ENCODING]
```

options:

```
-h, --help          show this help message and exit
--inpgnfn INPGNFN   Write the input pgn filename, required.
--outpgnfn OUTPGNFN Write the output pgn filename, required, mode=overwrite.
--sort-tag SORT_TAG Sort the games by tag. [default=eco, value=(eco | ecot | event |
↳date | round | white | black | site | plycount)]. e.g. --sort-tag event
--sort-direction SORT_DIRECTION
Write the direction to sort the games. [default=lowtohigh,
↳value=(lowtohigh | hightolow)].
--encoding ENCODING Encoding used in reading pgn file when sorting, not required.
↳[default=utf-8, value=(utf-8 | ISO-8859-1)]. If you encounter an error like
"UnicodeDecodeError: utf-8 codec cannot decode ..." you can try,
↳--encoding ISO-8859-1
```

Add eco help

```
pgnhelper addeco --help
```

```
usage: pgnhelper addeco [-h] --inpgnfn INPGNFN --outpgnfn OUTPGNFN --inecopgnfn
↳INECOPGNFN
```

options:

```
-h, --help          show this help message and exit
--inpgnfn INPGNFN   Write the input pgn filename, required.
--outpgnfn OUTPGNFN Write the output pgn filename, required, mode=overwrite.
--inecopgnfn INECOPGNFN
Write the reference eco.pgn filename, required.
```

Round-robin help

```
pgnhelper roundrobin --help
```

```
usage: pgnhelper roundrobin [-h] --inpgnfn INPGNFN --output OUTPUT [--win-point WIN_
↳POINT] [--draw-point DRAW_POINT] [--armageddon-file ARMAGEDDON_FILE]
[--win-point-arm WIN_POINT_ARM] [--loss-point-arm LOSS_POINT_
↳ARM] [--show-max-score] [--table-color TABLE_COLOR]
```

options:

```
-h, --help          show this help message and exit
--inpgnfn INPGNFN   Write the input pgn filename, required.
--output OUTPUT     Write the output filename, required, can be .html, .csv or .txt.
↳e.g --output tata_steel.html
--win-point WIN_POINT
The point when the player wins, default=1.0
--draw-point DRAW_POINT
The point when the player draws, default=0.5
--armageddon-file ARMAGEDDON_FILE
The armageddon pgn file, not required, default=None, if the
↳tournament is governed by armageddon tie-break system, you need to input the
armageddon pgn file.
```

(continues on next page)

(continued from previous page)

```

--win-point-arm WIN_POINT_ARM
    The point when one player wins the armageddon match, not
↳required, default=1.0
--loss-point-arm LOSS_POINT_ARM
    The point when one player loses the armageddon match, not
↳required, default=0.0
--show-max-score      A flag to show MaxScore column in the table, can be useful when
↳scoring is not standard.
--table-color TABLE_COLOR
    Write table color not required. [default="blue_light" value=(
↳"yellow_light", "grey_light", "orange_light", "green_light", "red_light",
    "yellow_dark", "grey_dark", "blue_dark", "orange_dark", "green_
↳dark", "red_dark")]

```

1.6.4 Usage

Command line

1. Add ECO

```

pgnhelper addeco --inpgnfn candidates_zurich_1953.pgn --outpgnfn eco_candidates_zurich_
↳1953.pgn --inecopgnfn eco.pgn

```

Note: You can get the [eco.pgn](#) from the pgnhelper github repository.

2. Sort

```

pgnhelper sort --inpgnfn sinqcup21.pgn --outpgnfn sorted_sinqcup21.pgn --sort-tag eco --
↳sort-direction hightolow

```

3. Generates a round-robin result table

```

pgnhelper roundrobin --inpgnfn sinqcup21.pgn --output sinqcup21.txt

```

Output with rating change and tie-break scores DE, Wins, SB and Koya:

Rank	Name	Rating	RChg	1	2	3	4	5	6	7	8	↳
↳9	10	Games	Score	Score%	DE	Wins	SB	Koya				
1	Vachier-Lagrave, Maxime	2751	13.74	x	0.5	0.0	0.5	0.5	1.0	1.0	0.5	1.
↳0	1.0	9	6.0	66.67	0.0	0	0.00	0.0				
2	Caruana, Fabiano	2806	1.03	0.5	x	0.5	0.5	0.5	1.0	0.0	1.0	0.
↳5	1.0	9	5.5	61.11	1.0	3	23.00	2.0				
3	Dominguez Perez, Leinier	2758	7.75	1.0	0.5	x	0.5	0.5	0.5	0.5	0.5	0.
↳5	1.0	9	5.5	61.11	1.0	2	24.00	2.5				
4	So, Wesley	2772	5.77	0.5	0.5	0.5	x	0.5	0.5	0.5	0.5	1.
↳0	1.0	9	5.5	61.11	1.0	2	22.75	2.0				
5	Rappport, Richard	2763	-2.96	0.5	0.5	0.5	0.5	x	0.5	0.5	0.0	1.
↳0	0.5	9	4.5	50.00	0.0	0	0.00	0.0				
6	Shankland, Sam	2709	-0.32	0.0	0.0	0.5	0.5	0.5	x	0.5	1.0	0.
↳5	0.5	9	4.0	44.44	1.5	1	16.75	1.5				

(continues on next page)

(continued from previous page)

7	Xiong, Jeffery	2710	-0.46	0.0	1.0	0.5	0.5	0.5	0.5	0.5	x	0.5	0.
↪5	0.0	9	4.0	44.44	1.0	1	19.00	2.5					
8	Mamedyarov, Shakhriyar	2782	-10.64	0.5	0.0	0.5	0.5	1.0	0.0	0.5	x	0.	
↪5	0.5	9	4.0	44.44	0.5	1	18.00	2.5					
9	Svidler, Peter	2714	-6.02	0.0	0.5	0.5	0.0	0.0	0.5	0.5	0.5	0.5	↪
↪x	1.0	9	3.5	38.89	0.0	0	0.00	0.0					
10	Swiercz, Dariusz	2655	-7.89	0.0	0.0	0.0	0.0	0.5	0.5	1.0	0.5	0.	
↪0	x	9	2.5	27.78	0.0	0	0.00	0.0					

4. Generates a round-robin table with armageddon tie-break as in Norway Chess

```
pgnhelper roundrobin --inpgnfn norwaychess.pgn --armageddon-file norwaychess_arm.pgn --
↪output norwaychess.html --win-point 3.0 --win-point-arm 1.5 --loss-point-arm 1.0 --
↪show-max-score
```

Rank	Name	Rating	RChg	1	2	3	4	5	6	7	8	9	↪
↪ 10	Games	Score	MaxScore	Score%	DE	Wins							
1	Carlsen, Magnus	2864	-0.05	x	3.0	1.0	1.5	1.0	3.0	1.5	1.0	3.0	↪
↪ 1.5	9	16.5	27.0	61.11	0.0	0							
2	Mamedyarov, Shakhriyar	2759	9.31	0.0	x	3.0	1.0	1.5	3.0	1.5	1.5	1.0	↪
↪ 3.0	9	15.5	27.0	57.41	0.0	0							
3	Anand, Viswanathan	2751	5.44	1.5	0.0	x	3.0	1.0	1.5	3.0	1.5	1.5	↪
↪ 1.5	9	14.5	27.0	53.70	0.0	0							
4	Vachier-Lagrave, Maxime	2750	5.58	1.0	1.5	0.0	x	3.0	1.5	1.5	3.0	1.0	↪
↪ 1.5	9	14.0	27.0	51.85	0.0	0							
5	So, Wesley	2776	-3.09	1.5	1.0	1.5	0.0	x	1.5	1.0	1.5	3.0	↪
↪ 1.5	9	12.5	27.0	46.30	0.0	0							
6	Giri, Anish	2761	-0.97	0.0	0.0	1.0	1.0	1.0	x	1.5	3.0	3.0	↪
↪ 1.5	9	12.0	27.0	44.44	0.0	0							
7	Topalov, Veselin	2730	-1.60	1.0	1.0	0.0	1.0	1.5	1.0	x	1.5	1.5	↪
↪ 1.0	9	9.5	27.0	35.19	1.5	0							
8	Tari, Aryan	2654	8.79	1.5	1.0	1.0	0.0	1.0	0.0	1.0	x	1.0	↪
↪ 3.0	9	9.5	27.0	35.19	1.0	1							
9	Radjabov, Teimour	2753	-14.84	0.0	1.5	1.0	1.5	0.0	0.0	1.0	1.5	x	↪
↪ 1.5	9	8.0	27.0	29.63	0.0	0							
10	Wang, Hao	2744	-8.57	1.0	0.0	1.0	1.0	1.0	1.0	1.5	0.0	1.0	↪
↪ x	9	7.5	27.0	27.78	0.0	0							

5. Generates swiss table

```
pgnhelper swiss --inpgnfn "./pgn/fide_grand_swiss_2021_riga.pgn" --output "fide_swiss.txt"
↪
# pgnhelper swiss --inpgnfn "./pgn/fide_grand_swiss_2021_riga.pgn" --output "fide_swiss.
↪html"
```

Rank	Name	Rating	RChg	R1	R2	R3	R4	R5	R6	↪			
↪R7	R8	R9	R10	R11	Games	Score	Score%	TB1	TB2	TB3	TB4	TB5	TB6
1	Firouzja, Alireza	2770	11.31	93W1	35B1	7W1	4B=	6W=	8B=	↪			
↪33W1	31W1	2B0	9W1	3B=	11	8.0	72.73	0.0	0.0	0.00	0.0	0	0
2	Caruana, Fabiano	2800	1.06	55W1	18B=	36W=	35B=	9W1	33B=	↪			
↪13W=	24B1	1W1	6B=	7W=	11	7.5	68.18	67.0	72.5	49.75	0.0	4	1

(continues on next page)

(continued from previous page)

3	Oparin, Grigoriy	2654	21.62	62W=	78B1	17W=	70B=	26W=	5B=	↵			
↵73W1	19B1	7W=	23B1	1W=	11	7.5	68.18	63.5	68.5	45.75	0.0	4	3
4	Yu, Yangyi	2704	9.20	5W=	63B1	61W1	1W=	49B=	13B=	↵			
↵15W=	28B=	33W1	8B=	6W=	11	7.0	63.64	66.5	72.0	44.50	0.0	3	1
5	Keymer, Vincent	2630	22.06	4B=	27W1	25B=	10W=	59B=	3W=	↵			
↵24B0	82W1	53B1	29W1	9B=	11	7.0	63.64	65.5	70.0	43.25	0.0	4	1
6	Vachier-Lagrave, Maxime	2763	2.12	52B=	51W1	71B=	60W1	1B=	32W1	↵			
↵31B=	8W=	11B=	2W=	4B=	11	7.0	63.64	65.0	70.0	43.50	0.0	3	0
7	Predke, Alexandr	2666	14.22	46B1	54W1	1B0	47W=	42B1	31W0	↵			
↵57B1	49W1	3B=	11W=	2B=	11	7.0	63.64	64.5	70.0	42.25	0.0	5	3
8	Shirov, Alexei	2659	16.99	43B=	87W1	50B=	92W1	36B1	1W=	↵			
↵14B=	6B=	23W=	4W=	10B=	11	7.0	63.64	64.5	68.5	41.50	0.0	3	1
9	Howell, David W L	2658	15.43	82B1	92W=	72B=	50W=	2B0	54W1	↵			
↵47B1	14W1	12W1	1B0	5W=	11	7.0	63.64	62.5	66.5	40.25	0.0	5	2
10	Sargissian, Gabriel	2664	10.56	72W=	83B=	94W1	5B=	70W1	15B=	↵			
↵18W=	34B=	28W1	13B=	8W=	11	7.0	63.64	61.5	65.5	40.50	0.0	3	0
11	Anton Guijarro, David	2658	13.25	37W=	70B=	62W1	17B0	98W1	91B1	↵			
↵19W=	21B1	6W=	7B=	13W=	11	7.0	63.64	61.0	65.0	39.25	0.0	4	2
12	Korobov, Anton	2690	6.07	48B=	20W1	47B=	42W=	50B1	49W=	↵			
↵17B=	41W1	9B0	34W=	31B1	11	7.0	63.64	60.5	66.0	41.50	0.0	4	2
13	Sevian, Samuel	2654	15.89	80B=	84W1	92B=	72W1	32B=	4W=	↵			
↵2B=	53W1	30B=	10W=	11B=	11	7.0	63.64	60.5	64.5	39.75	0.0	3	0
14	Esipenko, Andrey	2720	3.08	58B=	42W=	81B=	63W1	71B=	56W1	↵			
↵8W=	9B0	64B1	31W1	15B=	11	7.0	63.64	60.0	64.5	40.00	0.0	4	1
15	Deac, Bogdan-Daniel	2643	15.05	89B=	104W=	106B1	53W=	29B1	10W=	↵			
↵4B=	25W=	18B=	38B1	14W=	11	7.0	63.64	60.0	63.0	39.25	0.0	3	3
16	Artemiev, Vladislav	2699	4.40	76B1	50W=	75B=	31W=	51B=	47W=	↵			
↵48B=	52W=	49B1	17W=	34B1	11	7.0	63.64	56.5	61.5	39.00	0.0	3	3
...													

Tie-breaks:

TB1 = Buchholz Cut 1
 TB2 = Buchholz
 TB3 = Sonneborn-Berger
 TB4 = Direct Encounter
 TB5 = Number of wins
 TB6 = Number of wins as black

6. Generates standing table

Command line:

```

pgnhelper standing --inpgfn "./pgn/interzonal_1970_palma_de_mallorca.pgn" --output ↵
↵palma.txt

```

Output:

Rank	Name	Games	Score	Score%	DE	Wins	SB	Koya
1	Fischer R	23	18.5	80.43	0.0	0	0.00	0.0
2	Geller E	23	15.0	65.22	1.5	8	167.00	7.5
3	Larsen B	23	15.0	65.22	1.0	9	167.50	7.0

(continues on next page)

(continued from previous page)

4	Huebner R	23	15.0	65.22	0.5	10	155.25	5.0
5	Uhlmann W	23	14.0	60.87	0.5	10	141.50	5.5
6	Taimanov M	23	14.0	60.87	0.5	8	146.50	5.5
7	Portisch L	23	13.5	58.70	0.5	7	149.75	6.5
8	Smyslov V	23	13.5	58.70	0.5	7	141.00	5.5
9	Gligoric S	23	13.0	56.52	0.5	7	135.50	5.5
10	Polugaevsky L	23	13.0	56.52	0.5	5	146.75	6.5
11	Mecking H	23	12.5	54.35	0.5	7	130.00	5.5
12	Panno O	23	12.5	54.35	0.5	6	130.75	4.5
13	Hort V	23	11.5	50.00	0.0	0	0.00	0.0
14	Ivkov B	23	10.5	45.65	0.0	0	0.00	0.0
15	Minic D	23	10.0	43.48	1.0	5	96.00	2.5
16	Suttles D	23	10.0	43.48	0.0	4	105.75	4.5
17	Reshevsky S	23	9.5	41.30	0.0	0	0.00	0.0
18	Addison W	23	9.0	39.13	0.5	3	95.25	4.5
19	Matulovic M	23	9.0	39.13	0.5	2	98.50	4.5
20	Filip M	23	8.5	36.96	1.5	1	91.50	3.5
21	Ujtumen T	23	8.5	36.96	1.0	5	85.25	2.5
22	Naranja R	23	8.5	36.96	0.5	5	88.75	2.5
23	Rubinetti J	23	6.0	26.09	0.0	0	0.00	0.0
24	Jimenez Zerquera E	23	5.5	23.91	0.0	0	0.00	0.0

Script

1. Add ECO

```
import pgnhelper.app

a = pgnhelper.app.PgnHelper(
    'addeco',
    inpgnfn='mygames.pgn',
    outpgnfn='out_eco.pgn',
    inecopgnfn='eco.pgn')
a.start()
```

2. Sort games

```
import pgnhelper.app

a = pgnhelper.app.PgnHelper(
    'sort',
    inpgnfn='mygames.pgn',
    outpgnfn='out_sorted.pgn',
    sort_tag='eco',
    sort_direction='hightolow')
a.start()
```

3. Generate round-robin table

```
"""
The output can be a pandas dataframe, txt, csv and html.
```

(continues on next page)

(continued from previous page)

```

"""
import pgnhelper.roundrobin
import pgnhelper.utility

# Get the dataframe output.
rr = pgnhelper.roundrobin.RoundRobin(
    "airthings.pgn",
    winpoint=3.0, drawpoint=1.0)
df = rr.table()

# Print to console.
print(df.to_string(index=False))

# Save to html.
pgnhelper.utility.save(df, "airthings.html")

# Save to csv.
df.to_csv("airthings.csv", index=False)

```

4. Generate round-robin table with armageddon games as tie-break as in Norway Chess

```

"""
Generate a round-robin table and save results and html.
"""

import pgnhelper.roundrobin
import pgnhelper.utility

rr = pgnhelper.roundrobin.RoundRobin(
    "./pgn/norway_chess_2022_classical.pgn",
    infnarm="./pgn/norway_chess_2022_armageddon.pgn",
    winpoint=3.0,
    winpointarm=1.5,
    losspointarm=1.0)
df = rr.table()
pgnhelper.utility.save(df, "norway_chess.html")

```

5. Find the frequency of opening names played by players in a tournament

```

"""Get frequency of players that plays sicilian opening.
"""

import pgnhelper
import pandas as pd

opening_name = 'Sicilian'
pgnfn = 'wchcand22.pgn'

df, players, israting = pgnhelper.record.get_pgn_data(pgnfn)

```

(continues on next page)

(continued from previous page)

```

openings = df.Opening.unique()
# for o in openings:
#     print(o)

data = {}
for p in players:
    data1 = {}
    for o in openings:
        dfw = df.loc[(df.Opening == o) & (df.White == p)]
        dfb = df.loc[(df.Opening == o) & (df.Black == p)]
        wcnt = len(dfw)
        bcnt = len(dfb)
        total = wcnt + bcnt
        data1.update({o: {'w': wcnt, 'b': bcnt, 'total': total}})

    data.update({p: {'data': data1}})

mydata = []
for p in players:
    mydata.append([opening_name, p,
                  data[p]['data'][opening_name]['w'],
                  data[p]['data'][opening_name]['b'],
                  data[p]['data'][opening_name]['total']])

mydf = pd.DataFrame(
    mydata,
    columns=['Opening', 'Player', 'Wgames', 'Bgames', 'Total'])

mydf = mydf.sort_values(by=['Total', 'Wgames'], ascending=[False, False])
mydf = mydf.reset_index(drop=True)
print(mydf)

```

Output:

	Opening	Player	Wgames	Bgames	Total
0	Sicilian	Rapport, Richard	0	4	4
1	Sicilian	Caruana, Fabiano	3	0	3
2	Sicilian	Nakamura, Hikaru	3	0	3
3	Sicilian	Nepomniachtchi, Ian	2	0	2
4	Sicilian	Radjabov, Teimour	1	1	2
5	Sicilian	Duda, Jan-Krzysztof	0	2	2
6	Sicilian	Firouzja, Alireza	0	2	2
7	Sicilian	Ding, Liren	0	0	0

1.6.5 Pytest

Setup Guide

Copy the tests folder from the [pgnhelper](#) github repository. These folders have files that we can use to run pytest.

Typical directory structure:

```
c:/mypgnhelper
c:/mypgnhelper/tests/test_roundrobin.py
c:/mypgnhelper/tests/test_addeco.py
c:/mypgnhelper/tests/test_sortgames.py
```

Command line:

```
c:/pgnhelper> pytest
```

1.6.6 Change Log

Version 0.10.1 [2022-06-27]

1. Silence some flake8 warnings

Version 0.10.0 [2022-06-26]

1. Add command line in swiss table generation

Command line:

```
pgnhelper swiss --inpgnfn "./pgn/fide_grand_swiss_2021_riga.pgn" --output grandwiss2021.
↳txt --round 11
```

2. Add number of wins swiss tie-break

Tie breaks:

```
TB1 = Buchholz cut 1
TB2 = Buchholz
TB3 = Sonneborn-Berger
TB4 = Direct Encounter
TB5 = Most number of wins
```

Command line:

```
pgnhelper swiss --inpgnfn "./pgn/fide_grand_swiss_2021_riga.pgn" --output grand_swiss.
↳txt --round 11
```

Output:

Rank	R7	R8	R9	R10	R11	Name	Rating	RChg	R1	R2	R3	R4	R5	R6	
↳R7						Games	Score	Score%	TB1	TB2	TB3	TB4	TB5		↳
1						Firouzja, Alireza	2770	11.31	93W1	35B1	7W1	4B=	6W=	8B=	↳
↳33W1	31W1	2B0	9W1	3B=	11	8.0	72.73	0.0	0.0	0.00	0.0	0			
2						Caruana, Fabiano	2800	1.06	55W1	18B=	36W=	35B=	9W1	33B=	↳
↳13W=	23B1	1W1	6B=	7W=	11	7.5	68.18	67.0	72.5	49.75	0.0	4			

(continues on next page)

(continued from previous page)

3		Oparin, Grigoriy	2654	21.62	62W= 78B1 17W= 70B= 26W= 5B=	11	7.5	68.18	63.5	68.5	45.75	0.0	4	
↔73W1	19B1	7W= 24B1 1W=												
4		Yu, Yangyi	2704	9.20	5W= 63B1 61W1 1W= 49B= 13B=	11	7.0	63.64	66.5	72.0	44.50	0.0	3	
↔15W=	28B=	33W1 8B= 6W=												
5		Keymer, Vincent	2630	22.06	4B= 27W1 25B= 10W= 59B= 3W=	11	7.0	63.64	65.5	70.0	43.25	0.0	4	
↔23B0	82W1	53B1 29W1 9B=												
6		Vachier-Lagrave, Maxime	2763	2.12	52B= 51W1 71B= 60W1 1B= 32W1	11	7.0	63.64	65.0	70.0	43.50	0.0	3	
↔31B=	8W=	11B= 2W= 4B=												
7		Predke, Alexandr	2666	14.22	46B1 54W1 1B0 47W= 42B1 31W0	11	7.0	63.64	64.5	70.0	42.25	0.0	5	
↔57B1	49W1	3B= 11W= 2B=												
8		Shirov, Alexei	2659	16.99	43B= 87W1 50B= 92W1 36B1 1W=	11	7.0	63.64	64.5	68.5	41.50	0.0	3	
↔14B=	6B=	24W= 4W= 10B=												
9		Howell, David W L	2658	15.43	82B1 92W= 72B= 50W= 2B0 54W1	11	7.0	63.64	62.5	66.5	40.25	0.0	5	
↔47B1	14W1	12W1 1B0 5W=												
10		Sargissian, Gabriel	2664	10.56	72W= 83B= 94W1 5B= 70W1 15B=	11	7.0	63.64	61.5	65.5	40.50	0.0	3	
↔18W=	34B=	28W1 13B= 8W=												
11		Anton Guijarro, David	2658	13.25	37W= 70B= 62W1 17B0 98W1 91B1	11	7.0	63.64	61.0	65.0	39.25	0.0	4	
↔19W=	21B1	6W= 7B= 13W=												
12		Korobov, Anton	2690	6.07	48B= 20W1 47B= 42W= 50B1 49W=	11	7.0	63.64	60.5	66.0	41.50	0.0	4	
↔17B=	41W1	9B0 34W= 31B1												
13		Sevian, Samuel	2654	15.89	80B= 84W1 92B= 72W1 32B= 4W=	11	7.0	63.64	60.5	64.5	39.75	0.0	3	
↔2B=	53W1	30B= 10W= 11B=												
14		Esipenko, Andrey	2720	3.08	58B= 42W= 81B= 63W1 71B= 56W1	11	7.0	63.64	60.0	64.5	40.00	0.0	4	
↔8W=	9B0	64B1 31W1 15B=												
15		Deac, Bogdan-Daniel	2643	15.05	89B= 104W= 106B1 53W= 29B1 10W=	11	7.0	63.64	60.0	63.0	39.25	0.0	3	
↔4B=	25W=	18B= 38B1 14W=												
16		Artemiev, Vladislav	2699	4.40	76B1 50W= 75B= 31W= 51B= 47W=	11	7.0	63.64	56.5	61.5	39.00	0.0	3	
↔48B=	52W=	49B1 17W= 34B1												
...														

3. Add number of wins with black swiss tie-break

Tie breaks:

TB1 = Buchholz cut 1
 TB2 = Buchholz
 TB3 = Sonneborn-Berger
 TB4 = Direct Encounter
 TB5 = Most number of wins
 TB6 = Most number of wins **with** black

Output:

Rank	R8	R9	R10	R11	Name	Rating	RChg	R1	R2	R3	R4	R5	R6	
↔R7					Games	Score	Score%	TB1	TB2	TB3	TB4	TB5	TB6	
1					Firouzja, Alireza	2770	11.31	93W1	35B1	7W1	4B=	6W=	8B=	
↔33W1	31W1	2B0	9W1	3B=	11	8.0	72.73	0.0	0.0	0.00	0.0	0	0	
2					Caruana, Fabiano	2800	1.06	55W1	18B=	36W=	35B=	9W1	33B=	
↔13W=	24B1	1W1	6B=	7W=	11	7.5	68.18	67.0	72.5	49.75	0.0	4	1	
3					Oparin, Grigoriy	2654	21.62	62W=	78B1	17W=	70B=	26W=	5B=	
↔73W1	19B1	7W=	23B1	1W=	11	7.5	68.18	63.5	68.5	45.75	0.0	4	3	
4					Yu, Yangyi	2704	9.20	5W=	63B1	61W1	1W=	49B=	13B=	
↔15W=	28B=	33W1	8B=	6W=	11	7.0	63.64	66.5	72.0	44.50	0.0			

(continues on next page)

(continued from previous page)

5						Keymer, Vincent	2630	22.06	4B=	27W1	25B=	10W=	59B=	3W=	┌
↔24B0	82W1	53B1	29W1	9B=	11		7.0	63.64	65.5	70.0	43.25	0.0	4	1	
6						Vachier-Lagrave, Maxime	2763	2.12	52B=	51W1	71B=	60W1	1B=	32W1	┌
↔31B=	8W=	11B=	2W=	4B=	11		7.0	63.64	65.0	70.0	43.50	0.0	3	0	
7						Predke, Alexandr	2666	14.22	46B1	54W1	1B0	47W=	42B1	31W0	┌
↔57B1	49W1	3B=	11W=	2B=	11		7.0	63.64	64.5	70.0	42.25	0.0	5	3	
8						Shirov, Alexei	2659	16.99	43B=	87W1	50B=	92W1	36B1	1W=	┌
↔14B=	6B=	23W=	4W=	10B=	11		7.0	63.64	64.5	68.5	41.50	0.0	3	1	
9						Howell, David W L	2658	15.43	82B1	92W=	72B=	50W=	2B0	54W1	┌
↔47B1	14W1	12W1	1B0	5W=	11		7.0	63.64	62.5	66.5	40.25	0.0	5	2	
10						Sargissian, Gabriel	2664	10.56	72W=	83B=	94W1	5B=	70W1	15B=	┌
↔18W=	34B=	28W1	13B=	8W=	11		7.0	63.64	61.5	65.5	40.50	0.0	3	0	
11						Anton Guijarro, David	2658	13.25	37W=	70B=	62W1	17B0	98W1	91B1	┌
↔19W=	21B1	6W=	7B=	13W=	11		7.0	63.64	61.0	65.0	39.25	0.0	4	2	
12						Korobov, Anton	2690	6.07	48B=	20W1	47B=	42W=	50B1	49W=	┌
↔17B=	41W1	9B0	34W=	31B1	11		7.0	63.64	60.5	66.0	41.50	0.0	4	2	
13						Sevian, Samuel	2654	15.89	80B=	84W1	92B=	72W1	32B=	4W=	┌
↔2B=	53W1	30B=	10W=	11B=	11		7.0	63.64	60.5	64.5	39.75	0.0	3	0	
14						Esipenko, Andrey	2720	3.08	58B=	42W=	81B=	63W1	71B=	56W1	┌
↔8W=	9B0	64B1	31W1	15B=	11		7.0	63.64	60.0	64.5	40.00	0.0	4	1	
15						Deac, Bogdan-Daniel	2643	15.05	89B=	104W=	106B1	53W=	29B1	10W=	┌
↔4B=	25W=	18B=	38B1	14W=	11		7.0	63.64	60.0	63.0	39.25	0.0	3	3	
16						Artemiev, Vladislav	2699	4.40	76B1	50W=	75B=	31W=	51B=	47W=	┌
↔48B=	52W=	49B1	17W=	34B1	11		7.0	63.64	56.5	61.5	39.00	0.0	3	3	
...															

Version 0.9.0 [2022-06-25]

1. Generates swiss table

```
TB1 = Buchholz cut 1
TB2 = Buchholz
TB3 = Sonneborn-Berger
TB4 = Direct Encounter
```

Code

```
import pgnhelper.swiss
a = pgnhelper.swiss.Swiss("./pgn/fide_grand_swiss_2021_riga.pgn", round=11)
df = a.table()
print(df.to_string(index=False))
```

Output

Rank					Name	Rating	RChg	R1	R2	R3	R4	R5	R6	┌
↔R7	R8	R9	R10	R11	Games	Score	Score%	TB1	TB2	TB3	TB4			
1					Firouzja, Alireza	2770	11.31	93W1	35B1	7W1	4B=	6W=	8B=	┌
↔33W1	31W1	2B0	9W1	3B=	11	8.0	72.73	0.0	0.0	0.00	0.0			
2					Caruana, Fabiano	2800	1.06	55W1	18B=	36W=	35B=	9W1	33B=	┌
↔13W=	23B1	1W1	6B=	7W=	11	7.5	68.18	67.0	72.5	49.75	0.0			

(continues on next page)

(continued from previous page)

3	Oparin, Grigoriy					2654	21.62	62W=	78B1	17W=	70B=	26W=	5B=	␣
↔73W1	19B1	7W=	25B1	1W=	11	7.5	68.18	63.5	68.5	45.75	0.0			
4	Yu, Yangyi					2704	9.20	5W=	63B1	61W1	1W=	49B=	13B=	␣
↔15W=	28B=	33W1	8B=	6W=	11	7.0	63.64	66.5	72.0	44.50	0.0			
5	Keymer, Vincent					2630	22.06	4B=	27W1	24B=	10W=	59B=	3W=	␣
↔23B0	82W1	53B1	29W1	9B=	11	7.0	63.64	65.5	70.0	43.25	0.0			
6	Vachier-Lagrave, Maxime					2763	2.12	52B=	51W1	71B=	60W1	1B=	32W1	␣
↔31B=	8W=	11B=	2W=	4B=	11	7.0	63.64	65.0	70.0	43.50	0.0			
7	Predke, Alexandr					2666	14.22	46B1	54W1	1B0	47W=	42B1	31W0	␣
↔57B1	49W1	3B=	11W=	2B=	11	7.0	63.64	64.5	70.0	42.25	0.0			
8	Shirov, Alexei					2659	16.99	43B=	87W1	50B=	92W1	36B1	1W=	␣
↔14B=	6B=	25W=	4W=	10B=	11	7.0	63.64	64.5	68.5	41.50	0.0			
9	Howell, David W L					2658	15.43	82B1	92W=	72B=	50W=	2B0	54W1	␣
↔47B1	14W1	12W1	1B0	5W=	11	7.0	63.64	62.5	66.5	40.25	0.0			
10	Sargissian, Gabriel					2664	10.56	72W=	83B=	94W1	5B=	70W1	15B=	␣
↔18W=	34B=	28W1	13B=	8W=	11	7.0	63.64	61.5	65.5	40.50	0.0			
11	Anton Guijarro, David					2658	13.25	37W=	70B=	62W1	17B0	98W1	91B1	␣
↔19W=	21B1	6W=	7B=	13W=	11	7.0	63.64	61.0	65.0	39.25	0.0			
12	Korobov, Anton					2690	6.07	48B=	20W1	47B=	42W=	50B1	49W=	␣
↔17B=	41W1	9B0	34W=	31B1	11	7.0	63.64	60.5	66.0	41.50	0.0			
13	Sevian, Samuel					2654	15.89	80B=	84W1	92B=	72W1	32B=	4W=	␣
↔2B=	53W1	30B=	10W=	11B=	11	7.0	63.64	60.5	64.5	39.75	0.0			
14	Esipenko, Andrey					2720	3.08	58B=	42W=	81B=	63W1	71B=	56W1	␣
↔8W=	9B0	64B1	31W1	15B=	11	7.0	63.64	60.0	64.5	40.00	0.0			
15	Deac, Bogdan-Daniel					2643	15.05	89B=	104W=	106B1	53W=	29B1	10W=	␣
↔4B=	24W=	18B=	38B1	14W=	11	7.0	63.64	60.0	63.0	39.25	0.0			
16	Artemiev, Vladislav					2699	4.40	76B1	50W=	75B=	31W=	51B=	47W=	␣
↔48B=	52W=	49B1	17W=	34B1	11	7.0	63.64	56.5	61.5	39.00	0.0			
17	Petrosyan, Manuel					2605	21.43	99B1	32W=	3B=	11W1	18W=	25B=	␣
↔12W=	22B=	38W=	16B=	21W=	11	6.5	59.09	66.5	70.5	40.75	0.0			
18	Nihal, Sarin					2652	11.86	101W1	2W=	65B1	32W=	17B=	53W=	␣
↔10B=	38B=	15W=	21B=	19W=	11	6.5	59.09	64.0	68.0	38.75	0.0			
19	Dubov, Daniil					2714	-0.21	42B=	58W=	86B1	71W=	56B=	36W1	␣
↔11B=	3W0	41B1	24W=	18B=	11	6.5	59.09	61.5	66.0	37.50	0.0			
20	Kuzubov, Yuriy					2624	16.45	44W=	12B0	27B=	104W1	66B1	24W=	␣
↔59B=	60W=	32B1	23W=	30B=	11	6.5	59.09	61.5	65.0	36.50	0.0			
21	Fedoseev, Vladimir3					2704	-0.22	47B=	39W1	49B=	75W=	31B0	76W1	␣
↔35B1	11W0	60B1	18W=	17B=	11	6.5	59.09	59.5	64.5	37.00	0.0			
22	Sjugirov, Sanan					2663	7.89	88B=	57W1	48B0	58W1	47B=	39W1	␣
↔49B=	17W=	29B=	30W=	25W=	11	6.5	59.09	59.5	64.0	37.75	0.0			
23	Grandelius, Nils					2662	5.95	70W=	37B=	45W=	80B1	91W=	55B=	␣
↔5W1	2W0	39B=	20B=	61W1	11	6.5	59.09	59.5	63.5	36.50	0.0			
24	Kryvoruchko, Yuriy					2686	3.99	74W1	73B=	5W=	91B=	55W=	20B=	␣
↔80W1	15B=	34W=	19B=	26W=	11	6.5	59.09	59.5	63.5	36.50	0.0			
25	Vitiugov, Nikita					2727	-2.98	91W=	81B=	93W=	34B1	28W=	17W=	␣
↔71B1	32W1	8B=	3W0	22B=	11	6.5	59.09	59.5	63.5	36.50	0.0			
26	Aronian, Levon					2782	-10.65	51B=	52W1	60B=	48W=	3B=	71W=	␣
↔41B0	39W=	76B1	56W1	24B=	11	6.5	59.09	58.5	63.5	36.75	0.0			
27	Xiong, Jeffery					2700	-0.78	63W=	5B0	20W=	85B1	93W1	35B=	␣
↔34W0	74B1	37W=	52B1	28W=	11	6.5	59.09	58.5	62.5	34.25	0.0			
28	Sarana, Alexey					2649	9.98	92B0	82W1	54B=	76W1	25B=	72W1	␣
↔53B=	4W=	10B0	57W1	27B=	11	6.5	59.09	58.0	62.0	35.50	0.0			

(continues on next page)

(continued from previous page)

29	Alekseenko, Kirill	2710	-1.98	34W1	36B0	95W=	74B1	15W0	70B=	┌
→94W1	51B1 22W= 5B0 39W1	11	6.5	59.09	58.0	62.0	34.25	0.0		
30	Harikrishna, Pentala	2719	-3.62	81W=	91B0	107W1	95B1	35W=	61B=	┌
→55W1	33B= 13W= 22B= 20W=	11	6.5	59.09	55.5	58.5	33.50	0.0		
31	Sasikiran, Krishnan	2640	10.11	104B=	89W1	38W=	16B=	21W1	7B1	┌
→6W=	1B0 40W1 14B0 12W0	11	6.0	54.55	66.0	69.5	35.75	0.0		
32	Ponkratov, Pavel	2659	3.01	45W1	17B=	73W1	18B=	13W=	6B0	┌
→74W1	25B0 20W0 63B1 37W=	11	6.0	54.55	62.5	67.5	34.50	0.0		
33	Najer, Evgeniy	2654	6.44	84B=	80W1	43B1	36W=	48B1	2W=	┌
→1B0	30W= 4B0 61W= 46B=	11	6.0	54.55	62.5	67.0	34.00	0.0		
34	Shevchenko, Kirill	2632	8.98	29B0	105W1	66B=	25W0	87B1	65W1	┌
→27B1	10W= 24B= 12B= 16W0	11	6.0	54.55	62.5	66.0	33.00	0.0		
35	Swiercz, Dariusz	2647	8.88	103B1	1W0	88B1	2W=	30B=	27W=	┌
→21W0	73B= 50W1 37B= 40W=	11	6.0	54.55	62.0	66.0	32.75	0.0		
36	Saric, Ivan	2644	8.15	97W1	29W1	2B=	33B=	8W0	19B0	┌
→37W0	95B1 72W1 40B= 44W=	11	6.0	54.55	60.5	64.5	32.25	0.0		
37	Ter-Sahakyan, Samvel	2607	13.35	11B=	23W=	56B=	51W0	52B=	68W1	┌
→36B1	59W= 27B= 35W= 32B=	11	6.0	54.55	60.0	65.5	35.75	0.0		
38	Maghsoodloo, Parham	2701	-4.82	86B1	75W=	31B=	49W0	81B1	51W=	┌
→61B1	18W= 17B= 15W0 41B=	11	6.0	54.55	58.0	62.5	32.25	0.0		
39	Yilmaz, Mustafa	2626	10.92	40W=	21B0	98W=	107B1	44W1	22B0	┌
→91W1	26B= 23W= 59W1 29B0	11	6.0	54.55	58.0	61.0	30.00	0.0		
40	Eljanov, Pavel	2691	-4.50	39B=	47W0	85B=	46W1	54B=	50W1	┌
→51B=	48W1 31B0 36W= 35B=	11	6.0	54.55	57.5	62.0	33.75	0.0		
41	Volokitin, Andrei	2652	3.91	100B=	106W=	76B=	54W=	43B1	48W=	┌
→26W1	12B0 19W0 70B1 38W=	11	6.0	54.55	57.0	60.0	32.00	0.0		
42	Moussard, Jules	2632	6.43	19W=	14B=	79W1	12B=	7W0	80B0	┌
→67W=	94B= 97W= 84B1 77W1	11	6.0	54.55	56.0	60.0	31.50	0.0		
43	Henriquez Villagra, Cristobal	2608	12.27	8W=	69B1	33W0	61B=	41W0	93B=	┌
→96W=	91B= 79W1 55B1 53W=	11	6.0	54.55	54.0	58.0	31.00	0.0		
44	Van Foreest, Jorden	2691	-5.98	20B=	48W0	57B=	88W1	39B0	82W=	┌
→50B0	87W1 94B1 47W1 36B=	11	6.0	54.55	54.0	58.0	29.75	0.0		
45	Zvjaginsev, Vadim	2609	9.94	32B0	99W1	23B=	56W0	68B=	64W=	┌
→58B0	89W1 86B1 60W= 71B1	11	6.0	54.55	54.0	58.0	29.50	0.0		
46	Antipov, Mikhail Al.	2619	8.17	7W0	96B0	97W1	40B0	107W1	77B=	┌
→75W=	93B1 51W1 66B= 33W=	11	6.0	54.55	52.0	55.0	27.25	0.0		
47	Ponomariov, Ruslan	2631	3.77	21W=	40B1	12W=	7B=	22W=	16B=	┌
→9W0	65B= 70W= 44B0 80W1	11	5.5	50.00	63.5	68.0	32.75	0.0		
48	Hovhannisyan, Robert	2622	8.91	12W=	44B1	22W1	26B=	33W0	41B=	┌
→16W=	40B0 71B= 64W= 60B=	11	5.5	50.00	62.0	67.0	33.75	0.0		
49	Tari, Aryan	2646	-0.40	106B=	100W1	21W=	38B1	4W=	12B=	┌
→22W=	7B0 16W0 65B= 67W=	11	5.5	50.00	62.0	65.0	30.50	0.0		
50	Brkic, Ante	2621	6.68	77W1	16B=	8W=	9B=	12W0	40B0	┌
→44W1	71W= 35B0 97B1 66W=	11	5.5	50.00	61.5	65.5	30.75	0.0		
51	Tabatabaei, M. Amin	2639	3.99	26W=	6B0	100W1	37B1	16W=	38B=	┌
→40W=	29W0 46B0 73B1 65W=	11	5.5	50.00	61.5	65.5	30.50	0.0		
52	Niemann, Hans Moke	2638	2.24	6W=	26B0	67W=	78B=	37W=	98B1	┌
→62W1	16B= 59B= 27W0 57B=	11	5.5	50.00	60.0	64.0	30.25	0.0		
53	Navara, David	2691	-8.82	94W=	95B=	83W1	15B=	75W1	18B=	┌
→28W=	13B0 5W0 58B= 43B=	11	5.5	50.00	59.0	63.0	29.25	0.0		
54	Yakubboev, Nodirbek	2621	6.16	96W1	7B0	28W=	41B=	40W=	9B0	┌
→77W=	75B1 55W= 69B= 64B=	11	5.5	50.00	59.0	63.0	29.00	0.0		

(continues on next page)

(continued from previous page)

55	Chigaev, Maksim	2639	0.42	2B0	101W1	104B1	59W=	24B=	23W=	┐
→30B0	57W= 54B= 43W0 88B1	11	5.5	50.00	58.0	61.5	26.75	0.0		
56	Abdusattorov, Nodirbek	2646	-2.52	65W=	107B=	37W=	45B1	19W=	14B0	┐
→70W=	72B= 73W1 26B0 62W=	11	5.5	50.00	58.0	61.0	29.25	0.0		
57	Sadhvani, Raunak	2609	7.42	79W=	22B0	44W=	77B=	69W1	75B1	┐
→7W0	55B= 91W1 28B0 52W=	11	5.5	50.00	57.5	61.5	28.00	0.0		
58	Erigaisi, Arjun	2634	2.01	14W=	19B=	96W=	22B0	78W1	74B0	┐
→45W1	80B= 65W= 53W= 59B=	11	5.5	50.00	57.0	61.0	30.25	0.0		
59	Wojtaszek, Radoslaw	2691	-9.72	85W1	61B=	91W=	55B=	5W=	73B=	┐
→20W=	37B= 52W= 39B0 58W=	11	5.5	50.00	57.0	61.0	29.75	0.0		
60	Demchenko, Anton	2651	-0.72	67W=	98B1	26W=	6B0	73W0	83B1	┐
→95W1	20B= 21W0 45B= 48W=	11	5.5	50.00	57.0	61.0	27.50	0.0		
61	Bluebaum, Matthias	2640	-0.35	105B1	59W=	4B0	43W=	92B1	30W=	┐
→38W0	70B= 80W1 33B= 23B0	11	5.5	50.00	57.0	60.5	26.50	0.0		
62	Sindarov, Javokhir	2587	9.74	3B=	64W=	11B0	86W=	63B=	99W1	┐
→52B0	66W= 81B1 71W= 56B=	11	5.5	50.00	56.0	60.0	28.00	0.0		
63	Zhou, Jianchao	2629	0.70	27B=	4W0	89B1	14B0	62W=	84W1	┐
→82B=	64W0 88B1 32W0 87B1	11	5.5	50.00	55.5	60.0	26.25	0.0		
64	Cori, Jorge	2655	-5.46	78W=	62B=	70W0	67B=	83W=	45B=	┐
→85W1	63B1 14W0 48B= 54W=	11	5.5	50.00	55.0	59.5	28.75	0.0		
65	Kuybokarov, Temur	2549	14.55	56B=	102W1	18W0	93B=	77W=	34B0	┐
→81B1	47W= 58B= 49W= 51B=	11	5.5	50.00	53.5	57.5	26.75	0.0		
66	Svidler, Peter	2694	-11.53	95W=	94B=	34W=	73B=	20W0	85B=	┐
→72W=	62B= 74W1 46W= 50B=	11	5.5	50.00	53.0	57.0	27.75	0.0		
67	Nguyen, Thai Dai Van	2577	10.35	60B=	71W0	52B=	64W=	102B=	81W=	┐
→42B=	86W= 96B= 76W1 49B=	11	5.5	50.00	51.0	55.0	27.50	0.0		
68	Gelfand, Boris	2680	-11.18	73W0	74B=	78W=	83B=	45W=	37B0	┐
→84W=	97B= 95W1 72B= 85W1	11	5.5	50.00	49.5	53.5	25.50	0.0		
69	Rakhmanov, Aleksandr	2657	-10.61	87B=	43W0	84B=	106W=	57B0	89W=	┐
→92B=	101W1 82B= 54W= 83B1	11	5.5	50.00	47.5	50.5	23.75	0.0		
70	Indjic, Aleksandar	2612	1.76	23B=	11W=	64B1	3W=	10B0	29W=	┐
→56B=	61W= 47B= 41W0 81B=	11	5.0	45.45	62.5	67.0	29.75	0.0		
71	Donchenko, Alexander	2648	-2.80	98W=	67B1	6W=	19B=	14W=	26B=	┐
→25W0	50B= 48W= 62B= 45W0	11	5.0	45.45	61.5	65.5	29.25	0.0		
72	Praggnanandhaa, R	2618	1.80	10B=	90W1	9W=	13B0	96W1	28B0	┐
→66B=	56W= 36B0 68W= 75B=	11	5.0	45.45	59.5	63.5	26.25	0.0		
73	Kollars, Dmitrij	2621	0.10	68B1	24W=	32B0	66W=	60B1	59W=	┐
→3B0	35W= 56B0 51W0 97W1	11	5.0	45.45	59.0	63.0	26.75	0.0		
74	Onyshchuk, Volodymyr	2622	3.30	24B0	68W=	90B1	29W0	99B1	58W1	┐
→32B0	27W0 66B0 96W1 79W=	11	5.0	45.45	55.5	59.5	23.25	0.0		
75	Gukesh, D	2640	-6.89	108W1	38B=	16W=	21B=	53B0	57W0	┐
→46B=	54W0 100W= 101B1 72W=	11	5.0	45.45	55.0	57.0	23.25	0.0		
76	Pichot, Alan	2628	-4.92	16W0	108B1	41W=	28B0	101W1	21B0	┐
→97W=	96B1 26W0 67B0 100W1	11	5.0	45.45	54.0	56.0	19.00	0.0		
77	Matlakov, Maxim	2682	-16.05	50B0	85W=	87B=	57W=	65B=	46W=	┐
→54B=	88W= 83B= 82W1 42B0	11	5.0	45.45	52.0	56.5	24.75	0.0		
78	Goryachkina, Aleksandra	2602	0.56	64B=	3W0	68B=	52W=	58B0	90W=	┐
→102B=	79B0 103W= 92W1 99B1	11	5.0	45.45	51.0	55.0	22.50	0.0		
79	Cheparinov, Ivan	2659	-12.87	57B=	88W=	42B0	87W=	82B0	86W=	┐
→106B=	78W1 43B0 94W1 74B=	11	5.0	45.45	49.5	52.5	22.50	0.0		
80	Jobava, Baadur	2582	0.22	13W=	33B0	102B1	23W0	86B1	42W1	┐
→24B0	58W= 61B0 99W= 47B0	11	4.5	40.91	57.0	61.0	22.75	0.0		

(continues on next page)

(continued from previous page)

81	Kovalev, Vladislav	2634	-10.82	30B=	25W=	14W=	96B=	38W0	67B=	└
↔65W0	104B1 62W0 100B= 70W=	11	4.5	40.91	55.5	59.0	22.75	0.0		
82	Ivic, Velimir	2606	-8.46	9W0	28B0	105W=	108B1	79W1	44B=	└
↔63W=	5B0 69W= 77B0 89W=	11	4.5	40.91	55.5	57.5	19.50	0.0		
83	Sethuraman, S.P.	2620	-6.82	90B=	10W=	53B0	68W=	64B=	60W0	└
↔89B=	106W1 77W= 85B= 69W0	11	4.5	40.91	53.0	56.0	21.25	0.0		
84	Bartel, Mateusz	2597	-2.39	33W=	13B0	69W=	94B=	95W=	63B0	└
↔68B=	90W= 102B1 42W0 86B=	11	4.5	40.91	52.5	56.5	21.00	0.0		
85	Martirosyan, Haik M.	2624	-6.21	59B0	77B=	40W=	27W0	104B1	66W=	└
↔64B0	92W= 90B1 83W= 68B0	11	4.5	40.91	52.5	56.0	20.50	0.0		
86	Durabayli, Vasif	2629	-13.71	38W0	97B1	19W0	62B=	80W0	79B=	└
↔98W1	67B= 45W0 89B= 84W=	11	4.5	40.91	52.0	56.0	20.50	0.0		
87	Adly, Ahmed	2602	-5.95	69W=	8B0	77W=	79B=	34W0	97B0	└
↔103W1	44B0 98W1 93B1 63W0	11	4.5	40.91	52.0	56.0	19.75	0.0		
88	Ganguly, Surya Shekhar	2617	-6.01	22W=	79B=	35W0	44B0	105W1	96B=	└
↔93W=	77B= 63W0 91B1 55W0	11	4.5	40.91	51.5	55.0	19.75	0.0		
89	Vokhidov, Shamsiddin	2521	2.72	15W=	31B0	63W0	101B0	103W1	69B=	└
↔83W=	45B0 108B1 86W= 82B=	11	4.5	40.91	51.5	53.5	19.00	0.0		
90	Adhiban, Baskaran	2672	-24.55	83W=	72B0	74W0	98B0	108W1	78B=	└
↔101W=	84B= 85W0 103B= 104W1	11	4.5	40.91	44.0	46.0	16.50	0.0		
91	Dreev, Aleksey	2635	-12.35	25B=	30W1	59B=	24W=	23B=	11W0	└
↔39B0	43W= 57B0 88W0 92B=	11	4.0	36.36	60.5	64.5	24.00	0.0		
92	Neiksans, Arturs	2570	-4.08	28W1	9B=	13W=	8B0	61W0	94B0	└
↔69W=	85B= 93W= 78B0 91W=	11	4.0	36.36	56.0	60.0	22.50	0.0		
93	Abasov, Nijat	2638	-16.22	1B0	103W1	25B=	65W=	27B0	43W=	└
↔88B=	46W0 92B= 87W0 94B=	11	4.0	36.36	55.5	59.5	19.25	0.0		
94	Kravtsiv, Martyn	2625	-11.95	53B=	66W=	10B0	84W=	106B=	92W1	└
↔29B0	42W= 44W0 79B0 93W=	11	4.0	36.36	54.0	57.0	18.25	0.0		
95	Movsesian, Sergei	2627	-10.07	66B=	53W=	29B=	30W0	84B=	106W1	└
↔60B0	36W0 68B0 102W= 96B=	11	4.0	36.36	53.5	56.5	18.00	0.0		
96	Mamedov, Rauf	2673	-23.52	54B0	46W1	58B=	81W=	72B0	88W=	└
↔43B=	76W0 67W= 74B0 95W=	11	4.0	36.36	52.5	56.5	21.00	0.0		
97	Suleymanli, Aydin	2541	-3.63	36B0	86W0	46B0	105B=	100W1	87W1	└
↔76B=	68W= 42B= 50W0 73B0	11	4.0	36.36	52.0	55.5	18.50	0.0		
98	Bjerre, Jonas Buhl	2569	-5.46	71B=	60W0	39B=	90W1	11B0	52W0	└
↔86B0	99W= 87B0 107W1 102B=	11	4.0	36.36	50.5	53.5	17.00	0.0		
99	Jumabayev, Rinat	2658	-27.37	17W0	45B0	103W=	100B1	74W0	62B0	└
↔107W=	98B= 106W1 80B= 78W0	11	4.0	36.36	47.5	50.5	14.75	0.0		
100	Gretarsson, Hjorvar Steinn	2577	-10.37	41W=	49B0	51B0	99W0	97B0	108W1	└
↔105B1	102W= 75B= 81W= 76B0	11	4.0	36.36	47.0	49.0	15.25	0.0		
101	Georgiev, Kiril	2577	-12.63	18B0	55B0	108W=	89W1	76B0	102W=	└
↔90B=	69B0 104W1 75W0 107B=	11	4.0	36.36	47.0	49.0	14.75	0.0		
102	Paravyan, David	2642	-25.51	107W=	65B0	80W0	103B=	67W=	101B=	└
↔78W=	100B= 84W0 95B= 98W=	11	4.0	36.36	45.0	48.0	16.75	0.0		
103	Meshkovs, Nikita	2550	-8.96	35W0	93B0	99B=	102W=	89B0	105W=	└
↔87B0	107W1 78B= 90W= 108B=	11	4.0	36.36	43.0	45.0	14.50	0.0		
104	Miezis, Normunds	2467	-0.97	31W=	15B=	55W0	20B0	85W0	107B=	└
↔108B1	81W0 101B0 105W1 90B0	11	3.5	31.82	49.0	51.0	13.50	1.0		
105	Budisavljevic, Luka	2508	-11.93	61W0	34B0	82B=	97W=	88B0	103B=	└
↔100W0	108W= 107B= 104B0 106W1	11	3.5	31.82	42.0	44.0	11.75	0.0		
106	Van Foreest, Lucas	2543	-13.76	49W=	41B=	15W0	69B=	94W=	95B0	└
↔79W=	83B0 99B0 108W= 105B0	11	3.0	27.27	49.0	51.0	14.00	0.0		

(continues on next page)

(continued from previous page)

```

107      Morovic Fernandez, Ivan    2510 -11.77 102B= 56W= 30B0 39W0 46B0 104W=  1
↪99B= 103B0 105W= 98B0 101W=    11   3.0  27.27 47.5 51.0 12.25  0.0
108      Rakotomaharo, Fy Antenaina 2484 -21.75 75B0 76W0 101B= 82W0 90B0 100B0  1
↪104W0 105B= 89W0 106B= 103W=    11   2.0  18.18  0.0  0.0  0.00  0.0

```

2. Generates opening stats

Command line:

```
pgnhelper opening-stats --inpgnfn "./pgn/candidates_zurich_1953.pgn" --output candidates.
↪html
```

output options:
 candidates.txt
 candidates.csv

Code

```

>>> import pgnhelper.eco
>>> import pgnhelper.record
>>> df = pgnhelper.eco.get_opening_stats("./pgn/candidates_zurich_1953.pgn")
>>> df

```

Output

	Opening	Count	Count%
0	King's Indian Defence	44	20.95
1	Nimzo-Indian	41	19.52
2	Sicilian	23	10.95
3	English	18	8.57
4	Queen's Gambit Declined	16	7.62
5	Queen's Indian Defence	12	5.71
6	Ruy Lopez	10	4.76
7	Old Indian	7	3.33
8	French	6	2.86
9	Catalan	4	1.90
10	King's Indian	4	1.90
11	QGD Slav	4	1.90
12	Benoni	3	1.43
13	Dutch	3	1.43
14	Gruenfeld	3	1.43
15	QGA	3	1.43
16	Zukertort	3	1.43
17	Caro-Kann	2	0.95
18	Neo-Gruenfeld	2	0.95
19	Queen's pawn game	2	0.95

Note: Your game must have an opening info in the header.

Version 0.8.0

1. Add Koya system of breaking a tie in a round-robin tour.

```
--pgnhelper roundrobin --inpgnfn sinqcup21.pgn --output sinqcup21.txt
```

Rank	Name	Rating	RChg	1	2	3	4	5	6	7	8	Koya
↔9	10	Games	Score	Score%	DE	Wins	SB	Koya				
1	Vachier-Lagrave, Maxime	2751	13.74	x	0.5	0.0	0.5	0.5	1.0	1.0	0.5	1.
↔0	1.0	9	6.0	66.67	0.0	0	0.00	0.0				
2	Caruana, Fabiano	2806	1.03	0.5	x	0.5	0.5	0.5	1.0	0.0	1.0	0.
↔5	1.0	9	5.5	61.11	1.0	3	23.00	2.0				
3	Dominguez Perez, Leinier	2758	7.75	1.0	0.5	x	0.5	0.5	0.5	0.5	0.5	0.
↔5	1.0	9	5.5	61.11	1.0	2	24.00	2.5				
4	So, Wesley	2772	5.77	0.5	0.5	0.5	x	0.5	0.5	0.5	0.5	1.
↔0	1.0	9	5.5	61.11	1.0	2	22.75	2.0				
5	Rappport, Richard	2763	-2.96	0.5	0.5	0.5	0.5	x	0.5	0.5	0.0	1.
↔0	0.5	9	4.5	50.00	0.0	0	0.00	0.0				
6	Shankland, Sam	2709	-0.32	0.0	0.0	0.5	0.5	0.5	x	0.5	1.0	0.
↔5	0.5	9	4.0	44.44	1.5	1	16.75	1.5				
7	Xiong, Jeffery	2710	-0.46	0.0	1.0	0.5	0.5	0.5	0.5	x	0.5	0.
↔5	0.0	9	4.0	44.44	1.0	1	19.00	2.5				
8	Mamedyarov, Shakhriyar	2782	-10.64	0.5	0.0	0.5	0.5	1.0	0.0	0.5	x	0.
↔5	0.5	9	4.0	44.44	0.5	1	18.00	2.5				
9	Svidler, Peter	2714	-6.02	0.0	0.5	0.5	0.0	0.0	0.5	0.5	0.5	↔
↔x	1.0	9	3.5	38.89	0.0	0	0.00	0.0				
10	Swiercz, Dariusz	2655	-7.89	0.0	0.0	0.0	0.0	0.5	0.5	1.0	0.5	0.
↔0	x	9	2.5	27.78	0.0	0	0.00	0.0				

2. Add standing table generation.

```
--pgnhelper standing --inpgnfn interzonal_1970_palma_de_mallorca.pgn --output palma.txt
```

Rank	Name	Games	Score	Score%	DE	Wins	SB	Koya
1	Fischer R	23	18.5	80.43	0.0	0	0.00	0.0
2	Geller E	23	15.0	65.22	1.5	8	167.00	7.5
3	Larsen B	23	15.0	65.22	1.0	9	167.50	7.0
4	Huebner R	23	15.0	65.22	0.5	10	155.25	5.0
5	Uhlmann W	23	14.0	60.87	0.5	10	141.50	5.5
6	Taimanov M	23	14.0	60.87	0.5	8	146.50	5.5
7	Portisch L	23	13.5	58.70	0.5	7	149.75	6.5
8	Smyslov V	23	13.5	58.70	0.5	7	141.00	5.5
9	Gligoric S	23	13.0	56.52	0.5	7	135.50	5.5
10	Polugaevsky L	23	13.0	56.52	0.5	5	146.75	6.5
11	Mecking H	23	12.5	54.35	0.5	7	130.00	5.5
12	Panno O	23	12.5	54.35	0.5	6	130.75	4.5
13	Hort V	23	11.5	50.00	0.0	0	0.00	0.0
14	Ivkov B	23	10.5	45.65	0.0	0	0.00	0.0
15	Minic D	23	10.0	43.48	1.0	5	96.00	2.5
16	Suttles D	23	10.0	43.48	0.0	4	105.75	4.5
17	Reshevsky S	23	9.5	41.30	0.0	0	0.00	0.0
18	Addison W	23	9.0	39.13	0.5	3	95.25	4.5
19	Matulovic M	23	9.0	39.13	0.5	2	98.50	4.5

(continues on next page)

(continued from previous page)

20	Filip M	23	8.5	36.96	1.5	1	91.50	3.5
21	Ujtumen T	23	8.5	36.96	1.0	5	85.25	2.5
22	Naranja R	23	8.5	36.96	0.5	5	88.75	2.5
23	Rubinetti J	23	6.0	26.09	0.0	0	0.00	0.0
24	Jimenez Zerquera E	23	5.5	23.91	0.0	0	0.00	0.0

2. Refactor roundrobin.
3. Add record module.
4. Add help.rst.

Version 0.7.0

- Add rating change column in the round-robin table.

Superbet classic 2022, Bucharest Romania:

Rank	Name	Rating	RChg	1	2	3	4	5	6	7	8	
↔9	10 Games	Score	Score%	DE	Wins	SB						
1	Aronian, Levon	2765	9.50	x	0.5	1.0	1.0	0.5	0.5	0.5	0.5	0.
↔5	0.5	9	5.5	61.11	1.5	2	24.75					
2	So, Wesley	2776	7.93	0.5	x	0.5	0.5	0.5	0.5	1.0	0.5	1.
↔0	0.5	9	5.5	61.11	1.0	2	23.50					
3	Vachier-Lagrave, Maxime	2750	11.64	0.0	0.5	x	0.5	1.0	0.5	0.5	1.0	0.
↔5	1.0	9	5.5	61.11	0.5	3	23.00					
4	Dominguez Perez, Leinier	2753	1.21	0.0	0.5	0.5	x	0.5	1.0	0.5	0.0	1.
↔0	0.5	9	4.5	50.00	1.5	2	19.50					
5	Caruana, Fabiano	2786	-3.49	0.5	0.5	0.0	0.5	x	0.5	0.5	0.5	1.
↔0	0.5	9	4.5	50.00	1.0	1	19.25					
6	Deac, Bogdan-Daniel	2671	12.62	0.5	0.5	0.5	0.0	0.5	x	0.5	0.5	0.
↔5	1.0	9	4.5	50.00	0.5	1	19.75					
7	Nepomniachtchi, Ian	2773	-6.64	0.5	0.0	0.5	0.5	0.5	0.5	x	1.0	0.
↔0	0.5	9	4.0	44.44	1.0	1	18.00					
8	Firouzja, Alireza	2804	-11.04	0.5	0.5	0.0	1.0	0.5	0.5	0.0	x	0.
↔5	0.5	9	4.0	44.44	0.0	1	18.00					
9	Mamedyarov, Shakhriyar	2759	-9.65	0.5	0.0	0.5	0.0	0.0	0.5	1.0	0.5	↔
↔x	0.5	9	3.5	38.89	0.5	1	15.50					
10	Rappport, Richard	2776	-12.07	0.5	0.5	0.0	0.5	0.5	0.0	0.5	0.5	0.
↔5	x	9	3.5	38.89	0.5	0	15.75					

Version 0.6.1

- Restructure package modules.
- Add documentation.

Version 0.6.0

- Fix Sonneborn-Berger (SB) column

1.6.7 Read The Docs

Requirements needed for building the docs.

requirements.txt:

```
Sphinx==5.0.2
sphinxcontrib-napoleon==0.7
sphinx-rtd-theme
pgnhelper
```

Location: docs/requirements.txt

Update docs

command line:

```
(venv) PS F:\Github\pgnhelper> cd docs
(venv) PS F:\Github\pgnhelper\docs> make html
```

1.6.8 Credits

python chess

- **site:**
 - <https://python-chess.readthedocs.io/en/latest/>
 - <https://github.com/niklasf/python-chess>
- **usage:**
 - adding eco
 - round-robin result table generation

pgn-extract

- **site:**
 - <https://www.cs.kent.ac.uk/people/staff/djb/pgn-extract/>
- **usage:**
 - The eco.pgn in eco folder is based from the eco.pgn from pgn-extract.
 - cleaning pgn files

weekinchess

- **site:**
 - <https://theweekinchess.com/>
- **usage:**
 - game collections

pgn mentor

- **site:**
 - <https://www.pgnmentor.com/files.html>
- **usage:**
 - game collections

pretty-html-table

- **site:**
 - <https://pypi.org/project/pretty-html-table/>
- **usage:**
 - generation of pretty html tables

mark weeks

- **site:**
 - <https://www.mark-weeks.com/chess/wcc-indx.htm>
- **usage:**
 - game collections

pandas

- **site:**
 - <https://pandas.pydata.org/>
- **usage:**
 - data manipulation

1.6.9 Links

- [Github pgnhelper](#)
- [Pypi pgnhelper](#)
- [Readthedocs pgnhelper](#)

1.6.10 API

App

Manages job requests from user through the command line.

```
class pgnhelper.app.PgnHelper(job: str, inpgnfn: Optional[str] = None, outpgnfn: Optional[str] = None, inecopgnfn: Optional[str] = None, sort_tag: str = 'eco', sort_direction: str = 'lowtohigh', output: Optional[str] = None, winpoint: float = 1.0, drawpoint: float = 0.5, tablecolor: str = 'blue_light', encoding: str = 'utf-8', armageddonfile: Optional[str] = None, winpointarm: float = 1.0, losspointarm: float = 0.0, showmaxscore: bool = False, round: int = 20)
```

Manages user options to execute the job.

job

The kind of job to be done, sort, addeco and roundrobin.

inpgnfn

The input pgn file or path and filename.

outpgnfn

The output pgn file or path and filename.

inecopgnfn

The eco.pgn that will be used in addeco job.

sort_tag

Used in sorting games.

sort_direction

The output sorting ordering, lowtohigh or hightolow.

start()

Run the type of job to be done.

It will sort the games, add eco, opening and variation names to the games or generate a round-robin result table.

Add ECO

Add eco, opening and variation names to the input pgn file.

The eco codes, opening and variations names are coming from the file eco.pgn that you have to supply to pgnhelper for this to work.

eco.pgn file sources:

- eco.pgn from [pgn-extract](#)
- eco.pgn from [my eco repository in github](#)

- eco.pgn from pgnhelper repository

Example:

```
>>> import pgnhelper.eco
>>> pgnhelper.eco.add_eco("./pgn/candidates_zurich_1953.pgn", "eco_cz.pgn", "./eco/eco.
↳pgn")
```

Example output from eco_cz.pgn:

```
[Event "ct"]
[Site "Zurich"]
[Date "1953.??.??"]
[Round "01"]
[White "Szabo L"]
[Black "Geller E"]
[Result "0-1"]
[ECO "A15"]
[ECOT "E02"]
[Opening "English"]
[OpeningT "Catalan"]
[Variation "Anglo-Indian"]
[VariationT "open, 5.Qa4"]

1. c4 Nf6 2. g3 e6 3. Bg2 d5 4. d4 dxc4 5. Qa4+ Nbd7 ...
```

Note there are ECOT, OpeningT, and VariationT, these are new tags where T refers to Transposition. The ECO A15 is the ECO based on the first 2 moves and ECOT E02 is the ECO after 12 moves.

`pgnhelper.eco.add_eco(inpgnfn: str, outpgnfn: str, inecopgnfn: str, ply: int = 4, maxply: int = 24)`

Add eco, opening and variation names to the pgn file.

Parameters

- **inpgnfn** – The input pgn file.
- **outpgnfn** – The output file.
- **inecopgnfn** – The eco.pgn file.
- **ply** – The game ply to start classifying the opening.
- **maxply** – The max game ply to stop classifying the opening.

`pgnhelper.eco.create_eco_db(inecopgnfn: str)`

Creates a dictionary of eco data.

Parameters

inecopgnfn – The eco.pgn file to be converted to a dictionary.

`pgnhelper.eco.get_opening_stats(fn: str, is_arm: bool = False)`

Generates a dataframe of opening stats.

Opening, counts Sicilian, 4, ...

Parameters

fn – The pgn filename.

Returns

A dataframe of Opening and count

Sort games

Sort games by game tags.

class `pgnhelper.sort.Game`

Manages sorting of games.

add_line(*line: str*)

Saves lines read from each game.

Parameters

line – A line read from the game.

`pgnhelper.sort.read_games`(*inpgnfn: str, encoding: str = 'utf-8'*) → List[*Game*]

Read games from input pgn file.

Parameters

- **inpgnfn** – The input pgn file.
- **encoding** – Encoding used in reading a file. If you encounter a `UnicodeDecodeError` we can use `ISO-8859-1` instead of `utf-8`.

Returns

A list of *Game* objects.

`pgnhelper.sort.save_games`(*games: List[Game], outpgnfn: str*)

Save the games to output file.

Parameters

- **games** – A list of games.
- **outpgnfn** – The output file.

`pgnhelper.sort.sort_games`(*inpgnfn: str, outpgnfn: str, sort_tag: str, sort_direction: str, encoding: str = 'utf-8'*) → None

Sort based on criteria and save the games.

Read the input pgn file, sort it and save the sorted games in output file. The input file is not changed.

Parameters

- **inpgnfn** – The input pgn file.
- **outpgnfn** – The output pgn file.
- **sort_tag** – The sorting criteria, can be event, site, date, round, white, black and eco.
- **sort_direction** – Direction can be hightolow or lowtohigh.
- **encoding** – Encoding used in reading the input file.

Roundrobin

Generates a round-robin result table.

It reads the input pgn file and generates a dataframe of round-robin table. It also add columns for tie-break scores for tied players.

Typical tie-break system that can be applied to a round-robin tournament according to FIDE.

13.16.2. Individual Round-Robin Tournaments:

- Direct encounter
- The greater number of wins, including forfeits
- Sonneborn-Berger
- Koya System (to be implemented in pgnhelper)

<https://handbook.fide.com/files/handbook/C02Standards.pdf>

```
class pgnhelper.roundrobin.RoundRobin(infn: str, infnarm: Optional[str] = None, winpoint: float = 1.0,
                                       drawpoint: float = 0.5, winpointarm: float = 1.0, losspointarm:
                                       float = 0.0, showmaxscore: bool = False)
```

Manages round-robin result table generation.

infn

The input pgn file.

infnarm

The input pgn file with armageddon games.

winpoint

The point for the winner.

drawpoint

The point when player draws.

winpointarm

The point for the winner in armageddon game.

losspointarm

The point for the loser in armageddon game.

games_per_encounter() → int

Counts the number of games per encounter excluding armageddon.

Returns

The number of games per encounter.

player_ranking() → DataFrame

Generates a dataframe of player ranking.

Returns

A pandas dataframe of players ranking.

standing() → DataFrame

Returns a dataframe of player standing.

The standing is sorted by score, with tie-breaks DE, Wins and SB.

table() → DataFrame

Generates a round-robin result table.

The table is sorted by DE or Direct Encounter, Number of Wins and SB (sonneborn-Berger)

Returns

A pandas dataframe of round-robin table.

Swiss

Generates a swiss result table.

It reads the input pgn file and generates a dataframe of swiss table. It also add columns for tie-break scores for tied players.

Typical tie-break system that can be applied to a swiss tournament according to FIDE.

13.16.4. Individual Swiss Tournaments where not all the ratings are consistent:

- Buchholz Cut 1
- Buchholz
- Sonneborn-Berger
- Cumulative system - Sum of Progressive Scores
- Direct encounter
- The greater number of wins including forfeits
- The greater number of wins with Black pieces

13.16.5. Individual Swiss Tournaments where all the ratings are consistent:

- Buchholz Cut 1
- Buchholz
- Direct encounter
- AROC
- The greater number of wins including forfeits
- The greater number of wins with Black pieces
- The greater number of games with Black (unplayed games shall be counted as played with White)
- Sonneborn-Berger

<https://handbook.fide.com/files/handbook/C02Standards.pdf>

Other reference: FIDE Chess.com Grand Swiss 2021

4. 8. 3. Tie-breaks If two (2) or more players score the same points, the tie is to be decided by the following criteria, in order of priority:

- a) Buchholz Cut 1;
- b) Buchholz;
- c) Sonneborn-Berger;
- d) Direct encounter between the players in tie;
- e) Drawing of lots.

All tie-breaks are calculated as described in C.02.13 of the FIDE Handbook.

Tie-break supported by this library:

TB1 = Buchholz Cut 1 TB2 = Buchholz TB3 = Sonneborn-Berger TB4 = Direct Encounter TB5 = Number of wins
TB6 = Number of wins as black

class `pgnhelper.swiss.Swiss`(*infn: str, round: int = 20*)

Manages swiss result table generation.

infn

The input pgn file.

infnarm

The input pgn file with armageddon games.

winpoint

The point for the winner.

drawpoint

The point when player draws.

winpointarm

The point for the winner in armageddon game.

losspointarm

The point for the loser in armageddon game.

round

The number of rounds.

convert_score(*score: float*)

Convert 1.0 to 1, 0.0 to 0 and 0.5 to =

get_opp_info(*opp_data: List, df_final: DataFrame, dfr: DataFrame, p: str*) → Tuple[List, bool]

Creates result data to build swiss table.

player_ranking() → DataFrame

Generates a dataframe of player ranking.

Returns

A pandas dataframe of players ranking.

table() → DataFrame

Generates a swiss result table.

The table is sorted by [score, buchholz cut 1, buchholz, sonneborn-berger, direct encounter].

Returns

A pandas dataframe of swiss table.

Record

Manages conversion of pgn file into a pandas dataframe.

Read the games in the pgn file. Each game will be converted to a row with header:

Round, White, Black, WElo, BElo, Result, Wpt, Bpt, Arm, Eco, Opening, WRChg, BRChg

Example:

```
PS F:\Github\pgnhelper> python
>>> import pgnhelper
>>> df, players, rating = pgnhelper.record.get_pgn_data("./pgn/wchcand22.pgn")
>>> df
Round      White      Black  WElo  ...  Eco
↳Opening  WRChg    BRChg
  1.1  Duda, Jan-Krzysztof  Rapport, Richard  2750  ...  B44      Sicilian
↳defence  0.201367 -0.201367
  1.2      Ding, Liren  Nepomniachtchi, Ian  2806  ...  A20      English
↳opening  -5.573116  5.573116
  1.3  Caruana, Fabiano  Nakamura, Hikaru  2783  ...  C65      Ruy
↳Lopez  4.669486 -4.669486
```

`pgnhelper.record.get_pgn_data(fn, is_arm: bool = False, k: int = 10) → Tuple[DataFrame, List, bool]`

Converts games to dataframe.

Parameters

- **fn** – The pgn filename.
- **is_arm** – If pgn file has armageddon games.
- **k** – The rating change k factor.

Returns

df, players, is_rating

Tiebreak

Generates tie-break points on tied players.

Tie-breaks supported:

- Direct Encounter
- Number of wins
- Sonneborn-Berger
- Koya system

`pgnhelper.tiebreak.direct_encounter(result_df: DataFrame, ranking_df: DataFrame, winpoint: float = 1.0, drawpoint: float = 0.5, winpointarm: float = 1.5, losspointarm: float = 1.0, label: str = 'DE') → DataFrame`

Creates a dataframe with DE column or direct encounter.

Requirement:

It is only applied when tied players have played each other. In round-robin format this can be applied automatically. But for swiss format, the tied players have to be checked.

`pgnhelper.tiebreak.koya_system`(*result_df: DataFrame, ranking_df: DataFrame, winpoint: float = 1.0, drawpoint: float = 0.5*) → DataFrame

Creates a dataframe with Koya column for Koya system score.

Koya system - the number of points achieved against all opponents who have achieved 50 % or more.

11.5.4.3, <https://handbook.fide.com/files/handbook/C02Standards.pdf>

Parameters

- **result_df** – A dataframe of [Round, White, Black, Result ...].
- **ranking_df** – A dataframe of standing, [Name, Games, Score].

Returns

A ranking dataframe with Koya column.

`pgnhelper.tiebreak.num_wins`(*result_df: DataFrame, ranking_df: DataFrame, label: str = 'Wins', bwins: bool = False*) → DataFrame

Creates a dataframe with Win column.

If a game has an armageddon tie-break, we will only count the number of wins based from the normal game only.

Parameters

- **result_df** – The result dataframe.
- **ranking_df** – Ranking of players based on score.
- **label** – The label or header of the resulting dataframe.
- **bwins** – If true then only count wins by black. If not count all wins.

Returns

A dataframe of ranking with Wins column for tie-break.

`pgnhelper.tiebreak.sonneborn_berger`(*result_df: DataFrame, ranking_df: DataFrame, gpe: int = 1, winpoint: float = 1.0, drawpoint: float = 0.5, label: str = 'SB'*) → DataFrame

Creates a dataframe with SB column for Sonneborn-Berger score.

Armageddon games currently are excluded in the calculation.

Parameters

- **result_df** – A dataframe of [Round, White, Black, Result].
- **ranking_df** – A dataframe of standing, [Name, Games, Score].
- **gpe** – games per encounter

Returns

A dataframe of round-robin result table.

`pgnhelper.tiebreak.tb_buchholz`(*record_df: DataFrame, rank_df: DataFrame, cut: int = 0, label: str = 'TB1'*) → DataFrame

Calculates buchholz score or sum of opponents score.

This tie-break system is only applied for a tournament with swiss format.

Parameters

- **record_df** – A dataframe of tournament games records.
- **rank_df** – A dataframe with player ranking, initially at ['Name, Games, Score]. Later ['Name, Games, Score, Buchholz, ... tie-break system]

- **cut** – Cut the player opponent score, if cut is 0 the default then no one will be cut, this is the normal buchholz. If this is 1 then the lowest score will be cut. If this is 2 then the last two lowest scores will be cut. If value is -1 this is median or cut the highest and lowest. If value is -2 then cut the 2 highest and 2 lowest.

Returns

A dataframe of name and buchholz score.

Elo

`pgnhelper.elo.add_rating_change(df: DataFrame, is_rating: bool, k: int = 10) → DataFrame`

Adds rating change columns to existing df.

Parameters

- **df** – A dataframe of players match records.
- **is_rating** – True if players have rating.
- **k** – The k factor.

Returns

A dataframe of game records with elo rating change columns. [Round, White, Black, WElo, BElo, Result, Wpt, Bpt, Arm, WRChg, BRChg]

`pgnhelper.elo.expected_score(rating_a: int, rating_b: int) → float`

Calculates the expected score of player_a against player_b.

Parameters

- **rating_a** – Rating of player_a
- **rating_b** – Rating of player_b

Returns

expected score of player_a

Example:

```
>>> import pgnhelper.elo
>>> white_elo = 2600
>>> black_elo = 2500
>>> score = pgnhelper.elo.expected_score(white_elo, black_elo)
>>> score
0.6400649998028851
```

`pgnhelper.elo.get_rating(df: DataFrame, p: str) → Union[int, str]`

Gets the rating of player p.

Parameters

- **df** – A dataframe of players match records.
- **p** – A player name.

Returns

rating of player p

`pgnhelper.elo.get_rating_change(df: DataFrame, p: str, k: int = 10) → float`

Gets the rating change of player p.

The given df has a rating change column for each side. This rating change column was calculated using a k factor with value 10. The k parameter if not 10 will be used to recalculate the rating change for the given player.

Armageddon games are not included in the returned rating change value.

Parameters

- **df** – A dataframe of players match records.
- **p** – A player named p.
- **k** – The rating change k factor.

Returns

rating change

Example 1, get the rating change of a player in the PGN file:

```
>>> import pgnhelper.record
>>> import pgnhelper.elo
>>> df, players, is_rating = pgnhelper.record.get_pgn_data("./pgn/superbet_classic_
↳2022_bucharest.pgn")
>>> players
['Firouzja, Alireza', 'Aronian, Levon', 'So, Wesley', 'Nepomniachtchi, Ian',
↳'Caruana, Fabiano', 'Vachier-Lagrave, Maxime', 'Deac, Bogdan-Daniel', 'Mamedyarov,
↳ Shakhriyar', 'Dominguez Perez, Leinier', 'Rapport, Richard']
>>> rc_levon = pgnhelper.elo.get_rating_change(df, "Aronian, Levon", k=10)
>>> rc_levon
9.496967974633389
```

Example 2, get the rating change of black player:

```
>>> import pgnhelper.elo
>>> white_rating = 2700
>>> black_rating = 2600
>>> black_point = 1
>>> white_point = 0
>>> expected_score = pgnhelper.elo.expected_score(black_rating, white_rating)
>>> k = 10
>>> rating_change = k * (black_point - expected_score)
>>> rating_change
6.400649998028851
```

Utility

Helps build the round-robin result table.

`pgnhelper.utility.get_encounter_score(df: DataFrame, p: str, op: str, winpoint: float = 1.0, drawpoint: float = 0.5, winpointarm: float = 1.0, losspointarm: float = 0.0) → List[float]`

Calculates the scores between the player p and op.

Parameters

- **df** – A pandas dataframe containing players match results.

- **p** – A player name.
- **op** – Opponent of player p.
- **winpoint** – The point when player wins.
- **drawpoint** – The point when player draws.
- **winpointarm** – The point when player wins in armageddon game.
- **losspointarm** – The point when player loses in armageddon game.

Returns

A list of score for p and op, score[<p score>, <op score>].

`pgnhelper.utility.save(df: DataFrame, fn: str, tablecolor: str = 'blue_light') → None`

Save the dataframe.

The output can be a csv, txt and html. :param df: A pandas dataframe. :param fn: The output filename. :param tablecolor: The table color for html output.

PYTHON MODULE INDEX

p

`pgnhelper.app`, 30
`pgnhelper.eco`, 30
`pgnhelper.elo`, 38
`pgnhelper.record`, 36
`pgnhelper.roundrobin`, 33
`pgnhelper.sort`, 32
`pgnhelper.swiss`, 34
`pgnhelper.tiebreak`, 36
`pgnhelper.utility`, 39

A

add_eco() (in module *pgnhelper.eco*), 31
 add_line() (*pgnhelper.sort.Game* method), 32
 add_rating_change() (in module *pgnhelper.elo*), 38

C

convert_score() (*pgnhelper.swiss.Swiss* method), 35
 create_eco_db() (in module *pgnhelper.eco*), 31

D

direct_encounter() (in module *pgnhelper.tiebreak*),
 36
 drawpoint (*pgnhelper.roundrobin.RoundRobin* at-
 tribute), 33
 drawpoint (*pgnhelper.swiss.Swiss* attribute), 35

E

expected_score() (in module *pgnhelper.elo*), 38

G

Game (class in *pgnhelper.sort*), 32
 games_per_encounter() (*pgn-
 helper.roundrobin.RoundRobin* method),
 33
 get_encounter_score() (in module *pgnhelper.utility*),
 39
 get_opening_stats() (in module *pgnhelper.eco*), 31
 get_opp_info() (*pgnhelper.swiss.Swiss* method), 35
 get_pgn_data() (in module *pgnhelper.record*), 36
 get_rating() (in module *pgnhelper.elo*), 38
 get_rating_change() (in module *pgnhelper.elo*), 38

I

inecopgnfn (*pgnhelper.app.PgnHelper* attribute), 30
 infn (*pgnhelper.roundrobin.RoundRobin* attribute), 33
 infn (*pgnhelper.swiss.Swiss* attribute), 35
 infnarm (*pgnhelper.roundrobin.RoundRobin* attribute),
 33
 infnarm (*pgnhelper.swiss.Swiss* attribute), 35
 inpgnfn (*pgnhelper.app.PgnHelper* attribute), 30

J

job (*pgnhelper.app.PgnHelper* attribute), 30

K

koya_system() (in module *pgnhelper.tiebreak*), 36

L

losspointarm (*pgnhelper.roundrobin.RoundRobin* at-
 tribute), 33
 losspointarm (*pgnhelper.swiss.Swiss* attribute), 35

M

module
 pgnhelper.app, 30
 pgnhelper.eco, 30
 pgnhelper.elo, 38
 pgnhelper.record, 36
 pgnhelper.roundrobin, 33
 pgnhelper.sort, 32
 pgnhelper.swiss, 34
 pgnhelper.tiebreak, 36
 pgnhelper.utility, 39

N

num_wins() (in module *pgnhelper.tiebreak*), 37

O

outpgnfn (*pgnhelper.app.PgnHelper* attribute), 30

P

PgnHelper (class in *pgnhelper.app*), 30
pgnhelper.app
 module, 30
pgnhelper.eco
 module, 30
pgnhelper.elo
 module, 38
pgnhelper.record
 module, 36
pgnhelper.roundrobin
 module, 33

`pgnhelper.sort`
 module, 32
`pgnhelper.swiss`
 module, 34
`pgnhelper.tiebreak`
 module, 36
`pgnhelper.utility`
 module, 39
`player_ranking()` (*pgnhelper.roundrobin.RoundRobin*
 method), 33
`player_ranking()` (*pgnhelper.swiss.Swiss method*), 35

R

`read_games()` (*in module pgnhelper.sort*), 32
`round` (*pgnhelper.swiss.Swiss attribute*), 35
`RoundRobin` (*class in pgnhelper.roundrobin*), 33

S

`save()` (*in module pgnhelper.utility*), 40
`save_games()` (*in module pgnhelper.sort*), 32
`sonneborn_berger()` (*in module pgnhelper.tiebreak*),
 37
`sort_direction` (*pgnhelper.app.PgnHelper attribute*),
 30
`sort_games()` (*in module pgnhelper.sort*), 32
`sort_tag` (*pgnhelper.app.PgnHelper attribute*), 30
`standing()` (*pgnhelper.roundrobin.RoundRobin*
 method), 33
`start()` (*pgnhelper.app.PgnHelper method*), 30
`Swiss` (*class in pgnhelper.swiss*), 35

T

`table()` (*pgnhelper.roundrobin.RoundRobin method*),
 33
`table()` (*pgnhelper.swiss.Swiss method*), 35
`tb_buchholz()` (*in module pgnhelper.tiebreak*), 37

W

`winpoint` (*pgnhelper.roundrobin.RoundRobin attribute*),
 33
`winpoint` (*pgnhelper.swiss.Swiss attribute*), 35
`winpointarm` (*pgnhelper.roundrobin.RoundRobin*
 attribute), 33
`winpointarm` (*pgnhelper.swiss.Swiss attribute*), 35