basic_exploration

August 18, 2025

1 Activity: Basic Exploration

1.1 Introduction

In this activity you will practice using exploration methods on a data set containing games of online chess*. This activity includes some or all of the following, not necessarily in this order: - Viewing the data - Finding the mean - Finding the median - Standard deviation - Aggregations - Grouping

*The data set is from Kaggle.

1.2 Note

This data set is larger than those used in previous activities. Please run the cell below which uses the `info()` method to get a sense of the data before you begin.

[2]:
```
import pandas as pd

df = pd.read_csv('chess_games.csv')
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20058 entries, 0 to 20057
Data columns (total 17 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   game_id            20058 non-null  int64
 1   rated              20058 non-null  bool
 2   turns              20058 non-null  int64
 3   victory_status     20058 non-null  object
 4   winner             20058 non-null  object
 5   time_increment     20058 non-null  object
 6   white_id           20058 non-null  object
 7   white_rating       20058 non-null  int64
 8   black_id           20058 non-null  object
 9   black_rating       20058 non-null  int64
 10  moves              20058 non-null  object
 11  opening_code       20058 non-null  object
 12  opening_moves      20058 non-null  int64
```

```
 13  opening_fullname   20058 non-null  object
 14  opening_shortname  20058 non-null  object
 15  opening_response   1207 non-null   object
 16  opening_variation  14398 non-null  object
dtypes: bool(1), int64(5), object(11)
memory usage: 2.5+ MB
```

Question 1 Create two `DataFrame` objects called `first_three` and `last_three` and assign the first and last three rows of the data set to them, respectively.

[3]:
```
import pandas as pd
import pandas as pd

df = pd.read_csv("chess_games.csv")

first_three = df.head(3)

last_three = df.tail(3)

print("First three rows:\n", first_three)
print("\nLast three rows:\n", last_three)
```

```
First three rows:
    game_id  rated  turns victory_status winner time_increment  white_id  \
0        1  False     13    Out of Time  White           15+2  bourgris
1        2   True     16         Resign  Black           5+10      a-00
2        3   True     61           Mate  White           5+10    ischia

   white_rating   black_id  black_rating  \
0          1500       a-00          1191
1          1322  skinnerua          1261
2          1496       a-00          1500

                                              moves opening_code  \
0  d4 d5 c4 c6 cxd5 e6 dxe6 fxe6 Nf3 Bb4+ Nc3 Ba5…          D10
1  d4 Nc6 e4 e5 f4 f6 dxe5 fxe5 fxe5 Nxe5 Qd4 Nc6…          B00
2  e4 e5 d3 d6 Be3 c6 Be2 b5 Nd2 a5 a4 c5 axb5 Nc…          C20

   opening_moves                        opening_fullname    opening_shortname  \
0              5        Slav Defense: Exchange Variation         Slav Defense
1              4  Nimzowitsch Defense: Kennedy Variation  Nimzowitsch Defense
2              3   King's Pawn Game: Leonardis Variation     King's Pawn Game

  opening_response    opening_variation
0              NaN   Exchange Variation
1              NaN    Kennedy Variation
2              NaN  Leonardis Variation
```

```
Last three rows:
        game_id  rated  turns victory_status winner time_increment  \
20055    20056   True     35           Mate  White           10+0
20056    20057   True    109         Resign  White           10+0
20057    20058   True     78           Mate  Black           10+0

           white_id  white_rating      black_id  black_rating  \
20055      jamboger          1219  schaaksmurf3          1286
20056  marcodisogno          1360      jamboger          1227
20057      jamboger          1235         ffbob          1339

                                                   moves opening_code  \
20055  d4 d5 Bf4 Nc6 e3 Nf6 c3 e6 Nf3 Be7 Bd3 O-O Nbd…          D00
20056  e4 d6 d4 Nf6 e5 dxe5 dxe5 Qxd1+ Kxd1 Nd5 c4 Nb…          B07
20057  d4 d5 Bf4 Na6 e3 e6 c3 Nf6 Nf3 Bd7 Nbd2 b5 Bd3…          D00

       opening_moves                opening_fullname  opening_shortname  \
20055              3  Queen's Pawn Game: Mason Attack  Queen's Pawn Game
20056              4                      Pirc Defense       Pirc Defense
20057              3  Queen's Pawn Game: Mason Attack  Queen's Pawn Game

      opening_response opening_variation
20055              NaN      Mason Attack
20056              NaN               NaN
20057              NaN      Mason Attack
```

[ ]:
```
# Question 1 Grading Checks

assert first_three.shape == (3, 17), 'Make sure that you chose only the first three rows.'
assert last_three.shape == (3, 17), 'Make sure that you chose only the last three rows.'
```

Question 2 Create two new `DataFrame` objects called `white_lower_rating` and `white_higher_rating` that are assigned the rows of data where the white player's rating is less than 1200 and greater than or equal to 1800, respectively.

[4]:
```
import pandas as pd
df = pd.read_csv("chess_games.csv")

white_lower_rating = df[df["white_rating"] < 1200]

white_higher_rating = df[df["white_rating"] >= 1800]

print("White players with rating < 1200:\n", white_lower_rating.head())
```

```
print("\nWhite players with rating >= 1800:\n", white_higher_rating.head())
```

```
White players with rating < 1200:
     game_id  rated  turns victory_status winner time_increment  \
14       15  False     31           Mate  White          15+15
15       16  False     43         Resign  Black          15+15
16       17  False     52         Resign  Black          15+15
17       18  False     66           Mate  Black          15+16
18       19  False    101         Resign  Black          15+15

             white_id  white_rating           black_id  black_rating  \
14  shivangithegenius          1094      sureka_akshat          1141
15      sureka_akshat          1141  shivangithegenius          1094
16  shivangithegenius          1094      sureka_akshat          1141
17      sureka_akshat          1141  shivangithegenius          1094
18  shivangithegenius          1094          slam_ment          1300

                                                moves opening_code  \
14  e4 e5 Nf3 Nc6 Bc4 Nf6 Nc3 Bc5 O-O O-O d3 Ne8 B…          C50
15  e4 e5 Nf3 Nc6 Bc4 Nf6 Ng5 Qe7 O-O Nxe4 Nxe4 h6…          C57
16  e4 e5 Nf3 Nc6 Bc4 Nf6 Nc3 Bc5 O-O O-O Ne1 d6 d…          C50
17  e4 e5 Nf3 Nc6 Bc4 Nf6 Nc3 Bc5 O-O O-O Ne1 Ne8 …          C50
18  e4 e5 Nf3 d6 Bc4 Be6 d3 Bxc4 dxc4 c5 O-O h6 Nc…          C41

    opening_moves                                opening_fullname  \
14              8                Four Knights Game: Italian Variation
15              7  Italian Game: Two Knights Defense |  Knight At…
16              8                Four Knights Game: Italian Variation
17              8                Four Knights Game: Italian Variation
18              5                                Philidor Defense #3

    opening_shortname opening_response    opening_variation
14  Four Knights Game              NaN    Italian Variation
15       Italian Game              NaN  Two Knights Defense
16  Four Knights Game              NaN    Italian Variation
17  Four Knights Game              NaN    Italian Variation
18   Philidor Defense              NaN                   #3

White players with rating >= 1800:
     game_id  rated  turns victory_status winner time_increment      white_id  \
26       27   True     43         Resign  White          10+10  gmcarlsen403
39       40  False     34         Resign  White          30+30       rajuppi
43       44  False     66         Resign  White          40+40       rajuppi
48       49  False     35           Mate  White          40+40       rajuppi
49       50  False     41           Mate  White          40+40       rajuppi

    white_rating           black_id  black_rating  \
```

```
26          1825  shivangithegenius          1079
39          2449  shivangithegenius           978
43          2443  shivangithegenius           978
48          2436  shivangithegenius           978
49          2436  shivangithegenius           978

                                                  moves opening_code  \
26  e4 e5 Nf3 Nc6 d4 exd4 Bc4 Bc5 c3 dxc3 Bxf7+ Kx…          C45
39  c4 e5 Nc3 Bc5 g3 Nc6 Bg2 Nf6 Nf3 b6 Nxe5 Bb7 N…          A21
43  e4 c5 Nf3 d6 d4 cxd4 Nxd4 Nf6 Nc3 g6 Be3 Bg7 f…          B76
48  e4 c5 Nf3 d6 d4 cxd4 Nxd4 Nf6 Nc3 g6 Be3 Bg7 f…          B77
49  e4 d5 exd5 Qxd5 Qf3 Qe5+ Ne2 Nf6 d4 Qd5 Qxd5 N…          B01

    opening_moves                                  opening_fullname  \
26              8                          Scotch Game: Haxo Gambit
39              3  English Opening: King's English Variation |  R…
43             17  Sicilian Defense: Dragon Variation |  Yugoslav…
48             17  Sicilian Defense: Dragon Variation |  Yugoslav…
49              4      Scandinavian Defense: Mieses-Kotroc Variation

       opening_shortname opening_response         opening_variation
26           Scotch Game              NaN                Haxo Gambit
39        English Opening             NaN  King's English Variation
43       Sicilian Defense             NaN           Dragon Variation
48       Sicilian Defense             NaN           Dragon Variation
49  Scandinavian Defense              NaN   Mieses-Kotroc Variation
```

[ ]:
```
# Question 2 Grading Checks

assert isinstance(white_lower_rating, pd.DataFrame), 'Make sure that you are␣
↪creating a DataFrame object called white_lower_rating.'
assert isinstance(white_higher_rating, pd.DataFrame), 'Make sure that you are␣
↪creating a DataFrame object called white_higher_rating.'
```

Question 3 Using the `black_rating` column, create a `DataFrame` object called `top_10_percent_black` which is assigned the top 10% of black players by rating. That is, the only rows of where the `black_rating` is higher than 90% of all the `black_rating` values.

[11]:
```
import pandas as pd
df = pd.read_csv("chess_games.csv")
cutoff = df["black_rating"].quantile(0.90)
top_10_percent_black = df.loc[df["black_rating"] >= cutoff]
print(top_10_percent_black.shape)
print(top_10_percent_black.head())
```

```
(2011, 17)
     game_id  rated  turns victory_status winner time_increment  \
```

```
7          8  False      9         Resign  Black          15+30
78        79   True     47         Resign  Black           15+5
203      204   True     28         Resign  Black            7+2
291      292   True     69           Draw   Draw           20+0
364      365   True     40         Resign  Black           10+0

               white_id  white_rating        black_id  black_rating  \
7    daniel_likes_chess          1413        soultego          2108
78             oldpaths          1579  iris_foxspring          2105
203             taranga          1289   hknight_chess          2108
291          nadanadora          1611   qaisahleyorof          2059
364   joselito-carnaval          1780    arjunshenoyk          2132

                                                  moves opening_code  \
7                         e4 Nc6 d4 e5 d5 Nce7 c3 Ng6 b4          B00
78   d4 d5 Bf4 Nf6 Nf3 e6 e3 Bd6 Bg3 O-O Bd3 b6 Nbd…          D02
203  e4 e5 Nf3 Nc6 Bb5 a6 Ba4 Nf6 Nc3 Be7 d4 exd4 N…          C77
291  d4 d5 Nf3 Nf6 c4 e6 Nc3 c6 e3 Bd6 Bd3 Nbd7 e4 …          D46
364  e4 c5 Nf3 d6 d4 cxd4 Nxd4 Nf6 Nc3 a6 a3 Nbd7 B…          B90

     opening_moves                                opening_fullname  \
7                5  Nimzowitsch Defense: Kennedy Variation |  Link…
78               5                   Queen's Pawn Game: London System
203              9    Ruy Lopez: Morphy Defense |  Tarrasch Variation
291             12                Semi-Slav Defense: Chigorin Defense
364             10                Sicilian Defense: Najdorf Variation

       opening_shortname opening_response  opening_variation
7    Nimzowitsch Defense              NaN  Kennedy Variation
78     Queen's Pawn Game              NaN      London System
203            Ruy Lopez              NaN     Morphy Defense
291    Semi-Slav Defense              NaN   Chigorin Defense
364     Sicilian Defense              NaN  Najdorf Variation
```

[ ]:
```
# Question 3 Grading Checks

assert top_10_percent_black.shape == (2011, 17), 'Make sure that you are␣
↪selecting the top 10% of black players by rating. Hint: Try using a␣
↪conditional statement to check which black_rating values are in the top 10%.'
```