Untitled

January 30, 2021

[52]:
```
import lichess.api
from lichess.format import PGN, SINGLE_PGN, PYCHESS
from io import *
import os
import glob
import pandas as pd
import numpy as np
import datetime
import berserk
import rpy2
from rpy2 import robjects as ro

#Import required libraries
```

[ ]:
```
client = berserk.Client()
usuario = input('Insert username: ')
```

[ ]:
```
user = lichess.api.user(usuario)
pgn = lichess.api.user_games(usuario,max=1000000, format=SINGLE_PGN)

#print(pgn)
```

[ ]:
```
games=open("games.txt","w+")
games.write(pgn)
games.close()
file=open('games.txt',"r",encoding="utf8") #read pgn
lines=file.readlines() #extract each line of the pgn
consol=open("chess_stats.txt","w+") #Create new txt doc to dump the processed␣
↪info
i=0 #start at 0
for line in range(len(lines)): #start loop
        try:
                if str(lines[i+14]).replace('\n','') == '': #write lines if␣
↪game has no elo difference
                        try:
                                consol.write(str(lines[i]).
↪replace('\n',',')+str(lines[i+1]).replace('\n',',')+str(lines[i+2]).
↪replace('\n',',')+
                                str(lines[i+3]).replace('\n',',')+
                                str(lines[i+4]).
↪replace('\n',',')+str(lines[i+5]).replace('\n',',')+
                                str(lines[i+6]).
↪replace('\n',',')+str(lines[i+7]).replace('\n',',')+str(lines[i+8]).
↪replace('\n',',')+
                                str(lines[i+9]).
↪replace('\n',',')+str(lines[i+10]).replace('\n',',')+str(lines[i+11]).
↪replace('\n',',')+
                                str(lines[i+12]).
↪replace('\n',',')+str(lines[i+13]).replace('\n',',')+str(lines[i+14]).
↪replace('\n',',')+
                                ','+','+str(lines[i+15]).
↪replace('\n',',')+','+','+'\n')
                                i+=18
                        except:
                                pass
                        finally:
                                pass
                else: #write lines if game has elo difference
                        try:
                                consol.write(str(lines[i]).
↪replace('\n',',')+str(lines[i+1]).replace('\n',',')+str(lines[i+2]).
↪replace('\n',',')+
                                str(lines[i+3]).replace('\n',',')+
                                str(lines[i+4]).
↪replace('\n',',')+str(lines[i+5]).replace('\n',',')+
                                str(lines[i+6]).
↪replace('\n',',')+str(lines[i+7]).replace('\n',',')+str(lines[i+8]).
↪replace('\n',',')+
                                str(lines[i+9]).
↪replace('\n',',')+str(lines[i+10]).replace('\n',',')+str(lines[i+11]).
↪replace('\n',',')+
                                str(lines[i+12]).
↪replace('\n',',')+str(lines[i+13]).replace('\n',',')+str(lines[i+14]).
↪replace('\n',',')+
                                str(lines[i+15]).
↪replace('\n',',')+str(lines[i+16]).replace('\n',',')+str(lines[i+17]).
↪replace('\n',',')+
                                str(lines[i+18]).
↪replace('\n',',')+str(lines[i+19]).replace('\n',',')+'\n')
                                i+=20
                        except:
                                pass
                        finally:
                                pass
        except:
                pass
        finally:
                pass
consol.close()
file.close()
os.remove('games.txt')
```

[ ]:

```
openings=pd.DataFrame({'ECO':
 ↪['A00','A01','A02','A03','A04','A05','A06','A07','A08','A09',
                              ␣
 ↪'A10','A11','A12','A13','A14','A15','A16','A17','A18','A19',
                              ␣
 ↪'A20','A21','A22','A23','A24','A25','A26','A27','A28','A29',
                              ␣
 ↪'A30','A31','A32','A33','A34','A35','A36','A37','A38','A39',
                              ␣
 ↪'A40','A41','A42','A43','A44','A45','A46','A47','A48','A49',
                              ␣
 ↪'A50','A51','A52','A53','A54','A55','A56','A57','A58','A59',
                              ␣
 ↪'A60','A61','A62','A63','A64','A65','A66','A67','A68','A69',
                              ␣
 ↪'A70','A71','A72','A73','A74','A75','A76','A77','A78','A79','A80','A81','A82','A83','A84','
                    'Opening':["Start position","Nimzowitsch-Larsen␣
 ↪Attack","Bird  1.f4",
                               "Bird: 1...d5","Reti  1.Nf3","Reti: 1...
 ↪Nf6","Reti: 1...d5",
                               "Reti: KIA","Reti: KIA 2...c5","Reti: 2.
 ↪c4","English  1.c4",
                               "English: Caro-Kann Defence","English:␣
 ↪Caro-Kann Defence",
                               "English: 1...e6","English: Neo-Catalan␣
 ↪Declined","English: Anglo-Indian",
```

```
[ ]: df=pd.read_csv('chess_stats.txt',header=None)
     df=df.rename(columns={0: "Event", 1: "Site", 2: "Date",3: "White",4: "Black",5:␣
      ↪"Result",6: "UTCDate",
                           7: "UTCTime",8: "WhiteElo",9: "BlackElo",10:␣
      ↪"WhiteRatingDiff",11: "BlackRatingDiff",
                             12: "Variant",13: "TimeControl",
                           14: "ECO",15: "Termination",16: "w",17: "Gameplay",18:␣
      ↪"x",19: "y",20:"z"})
     df=df.drop(['Site','w','x','y','z'], axis=1)
```

```
[ ]: df['Event'] = df['Event'].str[8:20].str.replace('"]','').astype('str')
     df['White'] = df['White'].str[8:80].str.replace('"]','').astype('str')
     df['Black'] = df['Black'].str[8:80].str.replace('"]','').astype('str')
     df['Result'] = df['Result'].str[9:20].str.replace('"]','').astype('str')
     df['Date'] = pd.to_datetime(df['UTCDate'].str[10:40].str.replace('"]','')+'␣
      ↪'+df['UTCTime'].str[10:40].str.replace('"]','')).apply(lambda dt: datetime.
      ↪datetime(dt.year, dt.month, dt.day, dt.hour,2*(dt.minute //2)))
     df['WhiteElo'] = pd.to_numeric(df['WhiteElo'].str[11:21].str.
      ↪replace('"]',''),errors='coerce')
     df['BlackElo'] = pd.to_numeric(df['BlackElo'].str[11:21].str.
      ↪replace('"]',''),errors='coerce')
     df['WhiteRatingDiff'] = pd.to_numeric(df['WhiteRatingDiff'].str[18:25].str.
      ↪replace('"]',''),errors='coerce')
     df['BlackRatingDiff'] = pd.to_numeric(df['BlackRatingDiff'].str[18:25].str.
      ↪replace('"]',''),errors='coerce')
     df['Variant'] = df['Variant'].str[10:30].str.replace('"]','').astype('str')
     df['TimeControl'] = df['TimeControl'].str[14:24].str.replace('"]','').
      ↪astype('str')
     df['ECO'] = df['ECO'].str[6:12].str.replace('"]','').astype('str')
     df['Termination'] = df['Termination'].str[14:40].str.replace('"]','').
      ↪astype('str')
     df['Gameplay'] = df['Gameplay'].str[11:1000].str.replace('"]','').astype('str')
     df['Points'] = np.select([(df['White'] == str(usuario)) & (df['Result'] ==␣
      ↪'1-0'),
                               (df['Black'] == str(usuario)) & (df['Result'] ==␣
      ↪'0-1'),
                               (df['Black'] == str(usuario)) & (df['Result'] ==␣
      ↪'1-0'),
                               (df['White'] == str(usuario)) & (df['Result'] ==␣
      ↪'0-1'),
                               (df['Result'] == '1/2-1/2')],
                              [1, 1,0,0, 0.5])
     df['WDL'] = np.select([(df['Points'] == 1),(df['Points'] == 0),(df['Points'] ==␣
      ↪0.5)], ["W", "L", "D"])
     df['Against'] = np.select([(df['White'] == str(usuario)) & (df['WhiteElo'] >␣
      ↪df['BlackElo']),
                              (df['Black'] == str(usuario)) & (df['BlackElo'] >␣
↪df['WhiteElo']),
                              (df['White'] == str(usuario)) & (df['WhiteElo'] <␣
↪df['BlackElo']),
                              (df['Black'] == str(usuario)) & (df['BlackElo'] <␣
↪df['WhiteElo'])],
                             ["Worst Player", "Worst Player","Better␣
↪Player","Better Player"])
df['ELO'] = np.select([(df['White'] == str(usuario)),(df['Black'] ==␣
↪str(usuario))],[df['WhiteElo'],df['BlackElo']])
df['Opponent'] = np.select([(df['White'] == str(usuario)),(df['Black'] ==␣
↪str(usuario))],[df['Black'],df['White']])
df['EloDif'] = np.select([(df['White'] == str(usuario)),(df['Black'] ==␣
↪str(usuario))],[df['WhiteRatingDiff'],df['BlackRatingDiff']])
df=df.drop(['UTCDate','UTCTime'], axis=1)
df=df.dropna(axis=0)
```

[7]:
```
country=pd.read_excel('opp_country.xlsx')
#country
```

[ ]:
```
df=df.merge(openings,how='left')
df=df.merge(country,how='left')
df['Country'].fillna("INT", inplace = True)
df.to_csv('chess_stats.txt')
```

[11]:
```
df.head()
```

[11]:
```
          Event                 Date         White        Black Result  \
0  Rated Bullet  2021-01-25 22:02:00    santificex  vladimir5119    1-0
1  Rated Bullet  2021-01-25 22:00:00  legendarcina    santificex    0-1
2  Rated Bullet  2021-01-25 21:58:00    santificex       FreeGa1    0-1
3  Rated Bullet  2021-01-25 21:56:00    santificex        FLOSAN    1-0
4  Rated Bullet  2021-01-25 21:52:00    wildercoto    santificex    0-1

   WhiteElo  BlackElo  WhiteRatingDiff  BlackRatingDiff   Variant  …  ECO  \
0      1748      1766                6               -6  Standard  …  A40
1      1665      1743               -6                5  Standard  …  B01
2      1748      1800               -5                4  Standard  …  E51
3      1743      1719                5               -5  Standard  …  A40
4      1710      1737               -5                6  Standard  …  B01

    Termination                                         Gameplay Points  \
0  Time forfeit   c3 d5 3. e4 Bf5 4. exf5 e6 5. Nd2 exf5 6. Ngf…    1.0
1  Time forfeit   exd5 Qxd5 3. c4 Qd8 4. d4 Nf6 5. Nc3 e6 6. Nf…    1.0
2        Normal   c4 e6 3. e3 Nf6 4. Nc3 Bb4 5. Nf3 O-O 6. Be2 …    0.0
3  Time forfeit   c4 exd4 3. Qxd4 c5 4. Qd1 Nc6 5. Nf3 Nf6 6. e…    1.0
```

```
4  Time forfeit   exd5 Qxd5 3. Nf3 Qd8 4. g3 Nf6 5. Bg2 e6 6. O…    1.0

  WDL        Against   ELO      Opponent EloDif  \
0   W  Better Player  1748  vladimir5119      6
1   W   Worst Player  1743  legendarcina      5
2   L  Better Player  1748       FreeGa1     -5
3   W   Worst Player  1743        FLOSAN      5
4   W   Worst Player  1737    wildercoto      6

                        Opening
0              Queen's Pawn Game
1  Scandinavian (Centre Counter)
2         Nimzo-Indian: 4.e3 O-O
3              Queen's Pawn Game
4  Scandinavian (Centre Counter)

[5 rows x 21 columns]
```

[38]:
```
%load_ext rpy2.ipython
```

[39]:
```
from rpy2.robjects.lib.ggplot2 import ggplot
```

[51]:
```
%%R -w 20 -h 13 -u in
library(ggplot2)
library(gridExtra)
library(readr)
#chess_stats <- read_csv("chess_stats.csv",col_types = cols(Date =␣
 ↪col_character()))
trend<-ggplot(df, aes(x=chess_stats$Date,␣
 ↪y=chess_stats$ELO,colour=chess_stats$Event)) + geom_point()␣
 ↪+geom_smooth(method=lm,formula = x~y)+
theme_void()+theme(legend.position= 'none')

bars<-ggplot(df, aes(x = chess_stats$WDL, y =␣
 ↪chess_stats$EloDif,col=chess_stats$WDL,fill=chess_stats$WDL)) +␣
 ↪geom_bar(stat = "identity")+
facet_grid(cols = vars(chess_stats$Termination))+theme_void()+theme(legend.
 ↪position= 'none',strip.text.x = element_text(size=0))
grid.arrange(trend,bars)
```