The Spike of Global Interest in Online Chess

By Mikhail Mesyatsev (Kim), December 10,2020

This is a Jupyterlite Notebook hosted on the Kim Data Science Professional Certificate.

Table of Contents

1. Introduction
2. Chess.com Dataset
2.1 Titled Players
2.2 Chess.com Country
2.3 Chess Kids
3. Methodology: Exploring & Analyzing
3.1 Number of Players
3.2 Chess.com Country Popularity
3.3 Chess Players World Map
3.4 Clustering Chess Openings

1. Introduction

Chess is one of the most ancient board games in history and it has been played by millions of people worldwide. During the 2020 year, online chess has gained popularity that we have not seen before and millions of new fans have joined online chess platforms.

This project aims to illuminate the current global interest in the online chessport platforms by finding the number of chess players worldwide, the most popular chess openings, and the most popular chess platform in each country. The ultimate goal of this project is to discover which countries contributing to chess players from different continents and illuminating the spike of chess opening from around the world.

Libraries

```
!mamba install -y
!mamba install -y -c conda-forge
!mamba install -y -c conda-forge -c bioconda
!mamba install -y -c conda-forge -c bioconda -c anaconda
!mamba install -y -c conda-forge -c bioconda -c anaconda -c plotly
!mamba install -y -c conda-forge -c bioconda -c anaconda -c plotly -c kaleido
!mamba install -y -c conda-forge -c bioconda -c anaconda -c plotly -c kaleido -c numpy
!mamba install -y -c conda-forge -c bioconda -c anaconda -c plotly -c kaleido -c numpy -c pandas
!mamba install -y -c conda-forge -c bioconda -c anaconda -c plotly -c kaleido -c numpy -c pandas -c matplotlib-base
!mamba install -y -c conda-forge -c bioconda -c anaconda -c plotly -c kaleido -c numpy -c pandas -c matplotlib-base -c seaborn
```

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.graph_objects as go
import kaleido
import ipywidgets as widgets
import ipywidgets.widgets.interactive_output
from ipywidgets.widgets.interactive import interactive
from ipywidgets.widgets.interactive_output import interactive_output
from ipywidgets.widgets.widget_selection.Dropdown import Dropdown
from ipywidgets.widgets.widget_selection.RadioButtons import RadioButtons
from ipywidgets.widgets.widget_bool.Checkbox import Checkbox
from ipywidgets.widgets.widget_layout.Layout import Layout
```

```
%matplotlib inline
```

2. Chess.com Dataset

The chess dataset and geological location information were crawled by:

- Chess.com API endpoints were to downloaded public data from Chess.com API.
- The API has different methods such as: getting player info, games, countries, clubs, professional players among other endpoints.
- Google Geocoding API from Google API to locate the coordinates of each country.
- This popular notebook website is details how the chess dataset were crawled: chessdataset.ipynb.

To ease the usage of the data, I will explore these 3 chess groups:

- 2.1 Titled Players
- 2.2 Chess.com Country
- 2.3 Chess Kids

All these commands:

2.1 Titled Players

The Titled Players Chess Dataset (TPD), stands chess titles to classify the strength of a player. Once awarded, TPD titles are for life. Titled players are the highest-rated chess players worldwide. They are the top-ranked by any player. However, these titles are not necessarily indicative of the current strongest chess players.

There are the Official TPD Titles and some of the popular titles are:

```
!curl -o titled_tuesday.csv https://raw.githubusercontent.com/kimbochen/chess-data-analysis/main/data/titled_tuesday/titled_tuesday.csv
```

Using Chess.com API the following dataset was created on December 08, 2020.

```
df = pd.read_csv('titled_tuesday.csv')
df.shape, df.columns, df.dtypes, df.nunique()
df.isnull().sum, df.isnull.shape[0].sum()
df.dtypes.value_counts(), df.select_dtypes(include=['object'])
df.sample(5), df.tail(5), df.head(5)
```

```
df[ df[ df[chessdot_players.fide.isin(fide_ids)].title.isin(titles)].country.isin(countries)].shape
df[ df[ df[chessdot_players.fide.isin(fide_ids)].title.isin(titles)].country.isin(countries)].sample(5)
```

```
df[ df[ df[chessdot_players.fide.isin(fide_ids)].title.isin(titles)].country.isin(countries)].to_csv('titled_tuesday.csv')
df[ df[ df[chessdot_players.fide.isin(fide_ids)].title.isin(titles)].country.isin(countries)].to_excel('titled_tuesday.xlsx')
df[ df[ df[chessdot_players.fide.isin(fide_ids)].title.isin(titles)].country.isin(countries)].to_parquet('titled_tuesday.parquet')
df[ df[ df[chessdot_players.fide.isin(fide_ids)].title.isin(titles)].country.isin(countries)].to_pickle('titled_tuesday.pkl')
df[ df[ df[chessdot_players.fide.isin(fide_ids)].title.isin(titles)].country.isin(countries)].to_hdf('titled_tuesday.h5', key='df')
df[ df[ df[chessdot_players.fide.isin(fide_ids)].title.isin(titles)].country.isin(countries)].to_feather('titled_tuesday.feather')
df[ df[ df[chessdot_players.fide.isin(fide_ids)].title.isin(titles)].country.isin(countries)].to_msgpack('titled_tuesday.msgpack')
df[ df[ df[chessdot_players.fide.isin(fide_ids)].title.isin(titles)].country.isin(countries)].to_stata('titled_tuesday.dta')
df[ df[ df[chessdot_players.fide.isin(fide_ids)].title.isin(titles)].country.isin(countries)].to_sql('titled_tuesday', conn)
df[ df[ df[chessdot_players.fide.isin(fide_ids)].title.isin(titles)].country.isin(countries)].to_clipboard()
df[ df[ df[chessdot_players.fide.isin(fide_ids)].title.isin(titles)].country.isin(countries)].to_latex()
df[ df[ df[chessdot_players.fide.isin(fide_ids)].title.isin(titles)].country.isin(countries)].to_markdown()
df[ df[ df[chessdot_players.fide.isin(fide_ids)].title.isin(titles)].country.isin(countries)].to_xarray()
df[ df[ df[chessdot_players.fide.isin(fide_ids)].title.isin(titles)].country.isin(countries)].to_xml()
df[ df[ df[chessdot_players.fide.isin(fide_ids)].title.isin(titles)].country.isin(countries)].to_csv('titled_tuesday.csv')
df[ df[ df[chessdot_players.fide.isin(fide_ids)].title.isin(titles)].country.isin(countries)].to_excel('titled_tuesday.xlsx')
df[ df[ df[chessdot_players.fide.isin(fide_ids)].title.isin(titles)].country.isin(countries)].to_parquet('titled_tuesday.parquet')
df[ df[ df[chessdot_players.fide.isin(fide_ids)].title.isin(titles)].country.isin(countries)].to_pickle('titled_tuesday.pkl')
df[ df[ df[chessdot_players.fide.isin(fide_ids)].title.isin(titles)].country.isin(countries)].to_hdf('titled_tuesday.h5', key='df')
df[ df[ df[chessdot_players.fide.isin(fide_ids)].title.isin(titles)].country.isin(countries)].to_feather('titled_tuesday.feather')
df[ df[ df[chessdot_players.fide.isin(fide_ids)].title.isin(titles)].country.isin(countries)].to_msgpack('titled_tuesday.msgpack')
df[ df[ df[chessdot_players.fide.isin(fide_ids)].title.isin(titles)].country.isin(countries)].to_stata('titled_tuesday.dta')
df[ df[ df[chessdot_players.fide.isin(fide_ids)].title.isin(titles)].country.isin(countries)].to_sql('titled_tuesday', conn)
df[ df[ df[chessdot_players.fide.isin(fide_ids)].title.isin(titles)].country.isin(countries)].to_clipboard()
df[ df[ df[chessdot_players.fide.isin(fide_ids)].title.isin(titles)].country.isin(countries)].to_latex()
df[ df[ df[chessdot_players.fide.isin(fide_ids)].title.isin(titles)].country.isin(countries)].to_markdown()
df[ df[ df[chessdot_players.fide.isin(fide_ids)].title.isin(titles)].country.isin(countries)].to_xarray()
df[ df[ df[chessdot_players.fide.isin(fide_ids)].title.isin(titles)].country.isin(countries)].to_xml()
```

2.2 Chess.com Country

Chess.com API

2.3 Chess Kids

Chess.com API

3. Methodology: Exploring & Analyzing

We will explore the chess dataset and group categorizes, aspects the number of players for each group, their countries and differences between each.

```
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].sample(5)
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].shape
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].dtypes
```

Selecting Players only in 2020

```
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_csv('chessdot_kids_2_plus_years.csv')
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_excel('chessdot_kids_2_plus_years.xlsx')
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_parquet('chessdot_kids_2_plus_years.parquet')
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_pickle('chessdot_kids_2_plus_years.pkl')
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_hdf('chessdot_kids_2_plus_years.h5', key='df')
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_feather('chessdot_kids_2_plus_years.feather')
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_msgpack('chessdot_kids_2_plus_years.msgpack')
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_stata('chessdot_kids_2_plus_years.dta')
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_sql('chessdot_kids_2_plus_years', conn)
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_clipboard()
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_latex()
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_markdown()
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_xarray()
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_xml()
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_csv('chessdot_kids_2_plus_years.csv')
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_excel('chessdot_kids_2_plus_years.xlsx')
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_parquet('chessdot_kids_2_plus_years.parquet')
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_pickle('chessdot_kids_2_plus_years.pkl')
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_hdf('chessdot_kids_2_plus_years.h5', key='df')
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_feather('chessdot_kids_2_plus_years.feather')
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_msgpack('chessdot_kids_2_plus_years.msgpack')
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_stata('chessdot_kids_2_plus_years.dta')
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_sql('chessdot_kids_2_plus_years', conn)
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_clipboard()
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_latex()
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_markdown()
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_xarray()
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_xml()
```

3.1 Number of Players

3.2 Chess.com Country Popularity

Chess.com Top 75/75 Kids Players from 172 Countries (8/12/2020)

```
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_csv('chessdot_kids_2_plus_years.csv')
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_excel('chessdot_kids_2_plus_years.xlsx')
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_parquet('chessdot_kids_2_plus_years.parquet')
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_pickle('chessdot_kids_2_plus_years.pkl')
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_hdf('chessdot_kids_2_plus_years.h5', key='df')
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_feather('chessdot_kids_2_plus_years.feather')
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_msgpack('chessdot_kids_2_plus_years.msgpack')
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_stata('chessdot_kids_2_plus_years.dta')
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_sql('chessdot_kids_2_plus_years', conn)
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_clipboard()
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_latex()
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_markdown()
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_xarray()
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_xml()
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_csv('chessdot_kids_2_plus_years.csv')
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_excel('chessdot_kids_2_plus_years.xlsx')
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_parquet('chessdot_kids_2_plus_years.parquet')
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_pickle('chessdot_kids_2_plus_years.pkl')
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_hdf('chessdot_kids_2_plus_years.h5', key='df')
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_feather('chessdot_kids_2_plus_years.feather')
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_msgpack('chessdot_kids_2_plus_years.msgpack')
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_stata('chessdot_kids_2_plus_years.dta')
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_sql('chessdot_kids_2_plus_years', conn)
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_clipboard()
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_latex()
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_markdown()
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_xarray()
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_xml()
```

Titled Players

Chess.com has 85/85 Titled Players from 208 Countries (8/12/2020)

```
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_csv('chessdot_kids_2_plus_years.csv')
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_excel('chessdot_kids_2_plus_years.xlsx')
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_parquet('chessdot_kids_2_plus_years.parquet')
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_pickle('chessdot_kids_2_plus_years.pkl')
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_hdf('chessdot_kids_2_plus_years.h5', key='df')
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_feather('chessdot_kids_2_plus_years.feather')
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_msgpack('chessdot_kids_2_plus_years.msgpack')
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_stata('chessdot_kids_2_plus_years.dta')
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_sql('chessdot_kids_2_plus_years', conn)
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_clipboard()
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_latex()
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_markdown()
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_xarray()
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_xml()
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_csv('chessdot_kids_2_plus_years.csv')
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_excel('chessdot_kids_2_plus_years.xlsx')
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_parquet('chessdot_kids_2_plus_years.parquet')
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_pickle('chessdot_kids_2_plus_years.pkl')
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_hdf('chessdot_kids_2_plus_years.h5', key='df')
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_feather('chessdot_kids_2_plus_years.feather')
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_msgpack('chessdot_kids_2_plus_years.msgpack')
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_stata('chessdot_kids_2_plus_years.dta')
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_sql('chessdot_kids_2_plus_years', conn)
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_clipboard()
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_latex()
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_markdown()
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_xarray()
df[ df[ df[chessdot_kids_2_plus_years].country.isin(countries)].sample(5)].to_xml()
```

3.3 Chess Players World Map

Chess.com Top 50 Countries out of 208 from 15,311 million Players & Chess.com (12/12/20)

Number of Players
Chess.com Top 50 Countries out of 208 from 15,311 million Players & Chess.com (12/12/20)
United States of America
Canada
Brazil
Mexico
Argentina
Chile
Colombia
Peru
Venezuela
Ecuador
Bolivia
Uruguay
Paraguay
Costa Rica
Guatemala
Dominican Republic
Puerto Rico
Panama
Honduras
El Salvador
Nicaragua
Belize
Cuba
Haiti
Jamaica
Trinidad and Tobago
Bahamas
Barbados
Saint Lucia
Grenada
Saint Vincent and the Grenadines
Antigua and Barbuda
Dominica
Saint Kitts and Nevis
Aruba
Curaçao
Sint Maarten (Dutch part)
Bonaire, Sint Eustatius and Saba
Anguilla
Cayman Islands
British Virgin Islands
Turks and Caicos Islands
Montserrat

please proceed the world. We are going to explore and cover the Winners and Loser of opening chapter in Wesscup by analyzing and visualizing the data collected from Chess.com.

Clustering Info

3.4.1 Tilted Players Clusters
3.4.2 Analyze Openings from Tilted Players
3.4.3 Queen's Gambit Effect
3.4.4 Most Common Blitz-Openings
3.4.5 Most Common Bullet-Openings
3.4.6 Most Common Blitz-Openings
3.4.7 Most Common Bullet-Openings
3.4.8 Tilted Players in Chess.com
3.4.9 Tilted Players in Chess.com
3.4.10 Top Openings (ECO) from Grandmasters
3.4.11 Top Openings (ECO) from Grandmasters

3.4.1 Tilted Players Clusters

Using Chess.com API the following dataset was created on December 31, 2020.

```
chess_dataset_path = "/content/drive/MyDrive/Colab Notebooks/Chess_Project/chess_dataset_12_31_2020.csv"
df = pd.read_csv(chess_dataset_path, sep='|')
df.shape, df.dtypes
```

```
((9997, 12),
 id                int64
 rated             bool
 time_class       object
 time_control     object
 white            object
 black            object
 last             int64
 fen              object
 pgn              object
 tournament       object
 tournament_count int64
 match            object
 dtype: object)
```

	id	rated	time class	time control	white	black	last	fen	pgn	tournament	tournament count	match
0	4144401	2020-07-31	Daily Chess	259200	1	1	1596227	rnbqkbnr/pppppppp/8/8/4P3/8/PPPP1PPP/RNBQKBNR b KQkq - 0 1	[Event "Live Chess"] [Site "Chess.com"] [Date "2020.07.31"] [White "Kaiser_Ivan"] [Black "Ivan_Ivanovich"] [Result "1-0"] [ECO "B00"] [WhiteElo "1"] [BlackElo "1"] [TimeControl "259200"] [Termination "Kaiser_Ivan wins by resignation"] 1. e4 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 1... e6 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 2. Nf3 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 2... d5 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 3. Nc3 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 3... Bb4 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 4. e5 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 4... c5 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 5. a3 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 5... Bxc3+ {[%clk 0:00:00.0] [%emt 0:00:00.0]} 6. bxc3 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 6... Ne7 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 7. Qg4 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 7... Qc7 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 8. Qxg7 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 8... Rg8 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 9. Qxh7 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 9... cxb2 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 10. Bd3 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 10... Bd7 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 11. Qxg8 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 11... Bxg8 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 12. Bxb2 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 12... d4 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 13. Ne2 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 13... Nf5 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 14. Bxf5 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 14... exf5 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 15. Nxd4 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 15... Bh6 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 16. Nxf5 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 16... Bg7 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 17. Rd1 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 17... Bxf5 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 18. exf5 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 18... Rg4 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 19. Bf4 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 19... Rd8 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 20. Qd2 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 20... Qe7 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 21. Qe2 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 21... Nd5 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 22. c4 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 22... Nb6 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 23. Be5 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 23... Kf8 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 24. Bxf6 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 24... Nxc4 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 25. Qc2 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 25... Qxf6 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 26. Qxc4+ {[%clk 0:00:00.0] [%emt 0:00:00.0]} 26... Kg8 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 27. Qxf6 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 27... Rxf6 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 28. Rxd8+ {[%clk 0:00:00.0] [%emt 0:00:00.0]} 28... Rf8 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 29. Rxf8+ {[%clk 0:00:00.0] [%emt 0:00:00.0]} 29... Kxf8 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 30. Rb1 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 30... Ke7 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 31. Rb7+ {[%clk 0:00:00.0] [%emt 0:00:00.0]} 31... Kf6 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 32. Rxa7 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 32... Rxf5 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 33. Ra6+ {[%clk 0:00:00.0] [%emt 0:00:00.0]} 33... Kf7 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 34. Ra7+ {[%clk 0:00:00.0] [%emt 0:00:00.0]} 34... Kf6 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 35. Ra6+ {[%clk 0:00:00.0] [%emt 0:00:00.0]} 35... Kf7 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 36. Ra7+ {[%clk 0:00:00.0] [%emt 0:00:00.0]} 36... Kf6 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 37. Ra6+ {[%clk 0:00:00.0] [%emt 0:00:00.0]} 37... Kf7 {[%clk 0:00:00.0] [%emt 0:00:00.0]} 38. Ra7+ {[%clk 0:00:00.0] [%emt 0:00:00.0]} 38... Kf6 {[%clk 0:00:00.0] [%emt 0:00:00.0]}

essentially to the tourists when with successfuul shareamalgam, chess could bring opportunities and change the course of many lives. But it did not.

References:

- Chess Wikipedia
- 10 Things Everyone Knows About Chess
- 10 Amazing Interesting Facts about Chess
- 101 Interesting and Unbelievable Chess Facts
- Chess Facts for Kids
- 101 The Queen's Gambit actors Cast Netflix

Latest News:

Go to the SLANG

- The Queen's Gambit Chess Board Wallet Online - Wallpaper Japan
- Chess Facts: Fun Facts About Chess Pieces, Chess Boards, The Game Chess Trivia
- Fun Facts about chess - The Washington Post