



GO!



PYTHON

▼ Spogliatoio | Python: Data on national teams around the world since 1993

Caderno de **Códigos**

Desenvolvedor [Enzo Schitini](#)

Tópicos

1. Introdução ao material
2. Análise exploratória
3. Algoritmo
4. Gráficos
5. Storytelling

Os dados foram fornecidos pelo Kaggle

▼ 1. Introdução ao material

1.1. .

▼ 1.2. Importando as funções

```
from functools import reduce
import time
from time import sleep
import matplotlib
import pandas as pd
import missingno as msno

from adige import ArquivoCSV
from adige import valore_abbreviato
```

2. Outros arquivos

▼ 3. Análise exploratória

**DATA SCIENCE
EXPLORATORY ANALYSIS**



3.1. As colunas

Coluna	Descrição
id	Número da conta

```
data = './base.csv'

df = pd.read_csv(data)

print(df.info())

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 23921 entries, 0 to 23920
Data columns (total 25 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   date                                23921 non-null  object
1   home_team                           23921 non-null  object
2   away_team                           23921 non-null  object
3   home_team_continent                 23921 non-null  object
4   away_team_continent                 23921 non-null  object
5   home_team_fifa_rank                 23921 non-null  int64
6   away_team_fifa_rank                 23921 non-null  int64
7   home_team_total_fifa_points         23921 non-null  int64
8   away_team_total_fifa_points         23921 non-null  int64
9   home_team_score                     23921 non-null  int64
10  away_team_score                     23921 non-null  int64
11  tournament                           23921 non-null  object
12  city                                23921 non-null  object
13  country                             23921 non-null  object
14  neutral_location                     23921 non-null  bool
15  shoot_out                           23921 non-null  object
16  home_team_result                     23921 non-null  object
17  home_team_goalkeeper_score          8379 non-null   float64
18  away_team_goalkeeper_score          8095 non-null   float64
19  home_team_mean_defense_score        7787 non-null   float64
20  home_team_mean_offense_score        8510 non-null   float64
21  home_team_mean_midfield_score       8162 non-null   float64
22  away_team_mean_defense_score        7564 non-null   float64
23  away_team_mean_offense_score        8312 non-null   float64
24  away_team_mean_midfield_score       7979 non-null   float64
dtypes: bool(1), float64(8), int64(6), object(10)
memory usage: 4.4+ MB
None

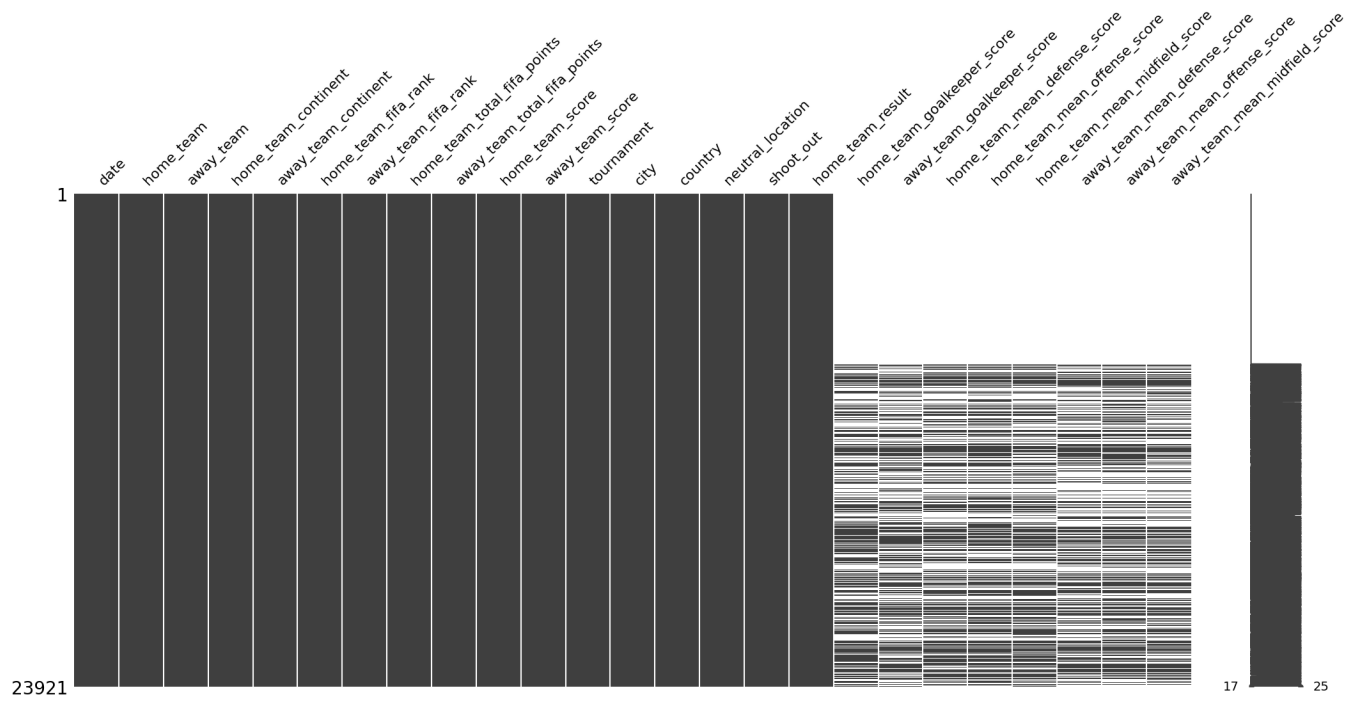
df.head(n=5)

   date   home_team  away_team  home_team_continent  away_team_continent  home_team_fifa_rank  away_team_fifa_rank  home_team_total_
0  1993-08-08    Bolivia    Uruguay          SouthAmerica          SouthAmerica              59              22
1  1993-08-08     Brazil     Mexico          SouthAmerica          NorthAmerica              8              14
2  1993-08-08    Ecuador  Venezuela          SouthAmerica          SouthAmerica             35              94
3  1993-08-08     Guinea  SierraLeone              Africa              Africa             65              86
4  1993-08-08    Paraguay    Argentina          SouthAmerica          SouthAmerica             67               5

5 rows x 25 columns

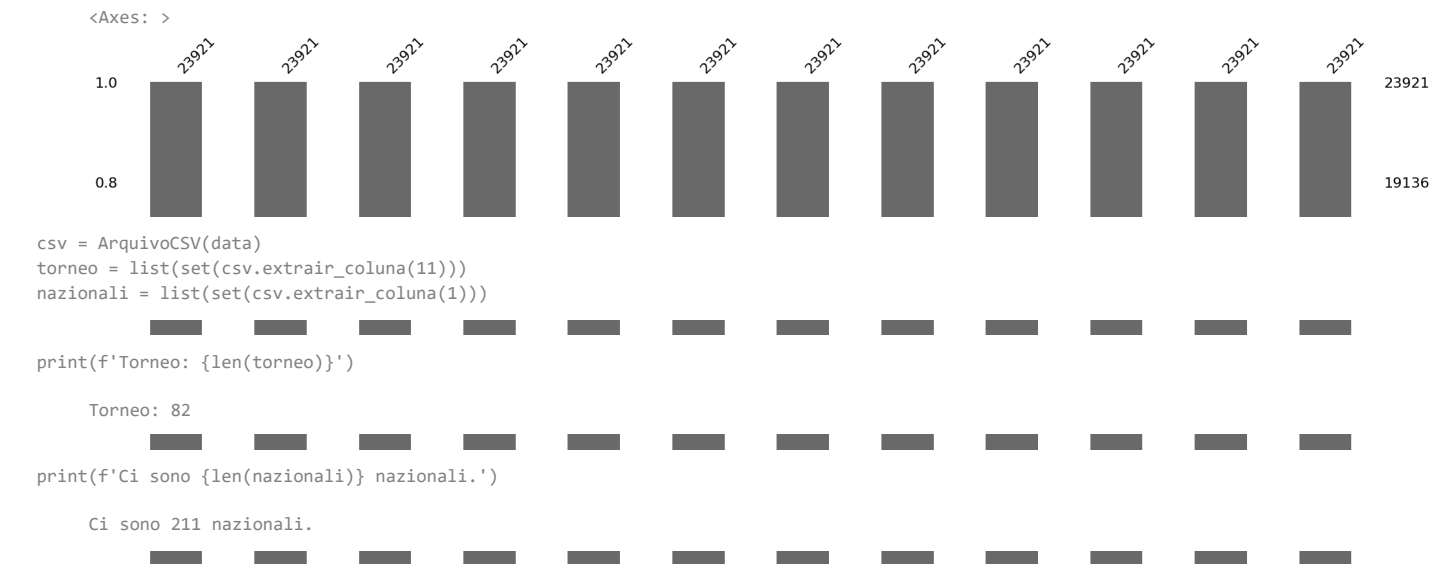
msno.matrix(df)
```

<Axes: >



```
df = df.drop(columns=["home_team_goalkeeper_score", "away_team_goalkeeper_score", "home_team_mean_defense_score", "home_team_mean_offense_score", "away_team_mean_defense_score", "away_team_mean_offense_score", "home_team_mean_midfield_score", "away_team_mean_midfield_score"])
df = df.dropna()

msno.bar(df)
```



3.2. Estrutura

```

line, colonne = df.shape

print(f'0 arquivo {data} tem:')
print(f'{valore_abbreviato(str(line))} linhas.')
print(f'{valore_abbreviato(str(colonne))} columnas.')

0 arquivo ./base.csv tem:
23.921 linhas.
12 columnas.

data = df

```

PYTHON ALGORITHM CALCULATION OF METRICS

4. Algoritmi e metriche

```

paese_da_cercare = 'Italy'

df_paese_totale = df[(df['home_team'] == paese_da_cercare) | (df['away_team'] == paese_da_cercare)]

print(f'Totale di partite: {valore_abbreviato(str(len(df_paese_totale)))}')

Totale di partite: 351

```

4.1. Prestazioni della squadra

- Vittorie, Sconfitte e Pareggi (Dentro e fuori casa)

```

class Prestazioni(object):

    import pandas as pd
    from adige import valore_abbreviato

    def __init__(self, paese: str, data) -> None:
        self.paese = paese
        self.data = data

```

```

def generale(self):
    df_paese_totale = self.data[(self.data['home_team'] == self.paese) | (self.data['away_team'] == self.paese)]
    totale = valore_abbreviato(str(len(df_paese_totale)))
    return totale

def vittorie(self):

    def calcoli(posto, score1, score2):
        df_goal = self.data[['home_team', 'away_team', 'home_team_score', 'away_team_score', 'tournament']]
        totale_di_partite = df_goal[df_goal[posto] == self.paese]
        vittorie = df_goal[(df_goal[posto] == self.paese) & (self.data[score1] > self.data[score2])]

        vittorie = valore_abbreviato(str(len(vittorie)))

        # Proporzione
        proporzione = round((float(vittorie) / len(totale_di_partite)) * 100, 2)
        return len(totale_di_partite), vittorie, proporzione

    partite_dentro_casa = calcoli('home_team', 'home_team_score', 'away_team_score')[0]
    vittorie_dentro_casa = calcoli('home_team', 'home_team_score', 'away_team_score')[1]
    proporzione_dentro_casa = calcoli('home_team', 'home_team_score', 'away_team_score')[2]

    partite_fuori_casa = calcoli('away_team', 'away_team_score', 'home_team_score')[0]
    vittorie_fuori_casa = calcoli('away_team', 'away_team_score', 'home_team_score')[1]
    proporzione_fuori_casa = calcoli('away_team', 'away_team_score', 'home_team_score')[2]

    return partite_dentro_casa, vittorie_dentro_casa, proporzione_dentro_casa, partite_fuori_casa, vittorie_fuori_casa, proporzione_f

def pareggi(self):

    def calcoli(posto, score1, score2):
        df_goal = self.data[['home_team', 'away_team', 'home_team_score', 'away_team_score', 'tournament']]
        totale_di_partite = df_goal[df_goal[posto] == self.paese]
        pareggi = df_goal[(df_goal[posto] == self.paese) & (self.data[score1] == self.data[score2])]

        pareggi = valore_abbreviato(str(len(pareggi)))

        # Proporzione
        proporzione = round((float(pareggi) / len(totale_di_partite)) * 100, 2)
        return len(totale_di_partite), pareggi, proporzione

    partite_dentro_casa = calcoli('home_team', 'home_team_score', 'away_team_score')[0]
    pareggi_dentro_casa = calcoli('home_team', 'home_team_score', 'away_team_score')[1]
    proporzione_dentro_casa = calcoli('home_team', 'home_team_score', 'away_team_score')[2]

    partite_fuori_casa = calcoli('away_team', 'away_team_score', 'home_team_score')[0]
    pareggi_fuori_casa = calcoli('away_team', 'away_team_score', 'home_team_score')[1]
    proporzione_fuori_casa = calcoli('away_team', 'away_team_score', 'home_team_score')[2]

    return partite_dentro_casa, pareggi_dentro_casa, proporzione_dentro_casa, partite_fuori_casa, pareggi_fuori_casa, proporzione_fuo

def sconfitte(self):

    def calcoli(posto, score1, score2):
        df_goal = self.data[['home_team', 'away_team', 'home_team_score', 'away_team_score', 'tournament']]
        totale_di_partite = df_goal[df_goal[posto] == self.paese]
        sconfitte = df_goal[(df_goal[posto] == self.paese) & (self.data[score1] < self.data[score2])]

        sconfitte = valore_abbreviato(str(len(sconfitte)))

        # Proporzione
        proporzione = round((float(sconfitte) / len(totale_di_partite)) * 100, 2)
        return len(totale_di_partite), sconfitte, proporzione

    partite_dentro_casa = calcoli('home_team', 'home_team_score', 'away_team_score')[0]
    sconfitte_dentro_casa = calcoli('home_team', 'home_team_score', 'away_team_score')[1]
    proporzione_dentro_casa = calcoli('home_team', 'home_team_score', 'away_team_score')[2]

    partite_fuori_casa = calcoli('away_team', 'away_team_score', 'home_team_score')[0]
    sconfitte_fuori_casa = calcoli('away_team', 'away_team_score', 'home_team_score')[1]
    proporzione_fuori_casa = calcoli('away_team', 'away_team_score', 'home_team_score')[2]

    return partite_dentro_casa, sconfitte_dentro_casa, proporzione_dentro_casa, partite_fuori_casa, sconfitte_fuori_casa, proporzione

italia = Prestazioni(paese_da_cercare, df)
print(f'{italia.paese} ha avuto {italia.generale()} partite.')

Italia ha avuto 351 partite.

# Dentro casa
print(f'{italia.vittorie()[0]} partite, {italia.vittorie()[1]} vittorie e {italia.vittorie()[2]}% di prestazione -- Dentro casa --')

```

```

# Fuori casa
print(f'{italia.vittorie()[3]} partite, {italia.vittorie()[4]} vittorie e {italia.vittorie()[5]}% di prestazione -- Fuori casa --')

199 partite, 117 vittorie e 58.79% di prestazione -- Dentro casa --
152 partite, 71 vittorie e 46.71% di prestazione -- Fuori casa --

# Dentro casa
print(f'{italia.pareggi()[0]} partite, {italia.pareggi()[1]} pareggi e {italia.pareggi()[2]}% di prestazione -- Dentro casa --')
# Fuori casa
print(f'{italia.pareggi()[3]} partite, {italia.pareggi()[4]} pareggi e {italia.pareggi()[5]}% di prestazione -- Fuori casa --')

199 partite, 55 pareggi e 27.64% di prestazione -- Dentro casa --
152 partite, 48 pareggi e 31.58% di prestazione -- Fuori casa --

# Dentro casa
print(f'{italia.sconfitte()[0]} partite, {italia.sconfitte()[1]} sconfitte e {italia.sconfitte()[2]}% di prestazione -- Dentro casa --')
# Fuori casa
print(f'{italia.sconfitte()[3]} partite, {italia.sconfitte()[4]} sconfitte e {italia.sconfitte()[5]}% di prestazione -- Fuori casa --')

199 partite, 27 sconfitte e 13.57% di prestazione -- Dentro casa --
152 partite, 33 sconfitte e 21.71% di prestazione -- Fuori casa --

```

▼ 4.2. Calcoli del torneo

```

def comportamento_in_un_torneo(paese, torneo:str, vedere: int):
    torneo_df = df[df['tournament'] == torneo]
    italia = Prestazioni(paese_da_cercare, torneo_df)
    print(f'{paese} nel torneo {torneo.upper()} ha avuto {italia.generale()} partite:')
    print('-----')

    lista = [italia.vittorie()[0], italia.vittorie()[1], italia.vittorie()[2], italia.vittorie()[3], italia.vittorie()[4], italia.vittorie()[5],
             italia.pareggi()[0], italia.pareggi()[1], italia.pareggi()[2], italia.pareggi()[3], italia.pareggi()[4], italia.pareggi()[5],
             italia.sconfitte()[0], italia.sconfitte()[1], italia.sconfitte()[2], italia.sconfitte()[3], italia.sconfitte()[4], italia.sconfitte()[5]]

    if vedere == 1:
        #
        print(f'{italia.vittorie()[0]} partite, {italia.vittorie()[1]} vittorie e {italia.vittorie()[2]}% di prestazione -- Dentro casa')
        print(f'{italia.vittorie()[3]} partite, {italia.vittorie()[4]} vittorie e {italia.vittorie()[5]}% di prestazione -- Fuori casa\n')

        print(f'{italia.pareggi()[0]} partite, {italia.pareggi()[1]} pareggi e {italia.pareggi()[2]}% di prestazione -- Dentro casa')
        print(f'{italia.pareggi()[3]} partite, {italia.pareggi()[4]} pareggi e {italia.pareggi()[5]}% di prestazione -- Fuori casa\n')

        print(f'{italia.sconfitte()[0]} partite, {italia.sconfitte()[1]} sconfitte e {italia.sconfitte()[2]}% di prestazione -- Dentro casa')
        print(f'{italia.sconfitte()[3]} partite, {italia.sconfitte()[4]} sconfitte e {italia.sconfitte()[5]}% di prestazione -- Fuori casa')

    return lista

# FIFAWorldCup Friendly CONCACAFNationsLeague UEFAEuroqualification FIFAWorldCupqualification CopaAmérica UEFAEuro
paese_da_cercare = 'Italy'
torneo_europa = ['FIFAWorldCup', 'Friendly', 'UEFAEuroqualification', 'FIFAWorldCupqualification', 'UEFAEuro', 'CopaAmérica']

for x in torneo_europa:
    try:
        comportamento_in_un_torneo(paese_da_cercare, x, 1)
    except ZeroDivisionError:
        print('\nATTENZIONE!')
        print(f'Mi sa che {paese_da_cercare} non ha mai partecipato al torneo: {x}.\n')

Italy nel torneo FIFAWORLDCUP ha avuto 29 partite:
-----
19 partite, 9 vittorie e 47.37% di prestazione -- Dentro casa
10 partite, 5 vittorie e 50.0% di prestazione -- Fuori casa

19 partite, 6 pareggi e 31.58% di prestazione -- Dentro casa
10 partite, 3 pareggi e 30.0% di prestazione -- Fuori casa

19 partite, 4 sconfitte e 21.05% di prestazione -- Dentro casa
10 partite, 2 sconfitte e 20.0% di prestazione -- Fuori casa
Italy nel torneo FRIENDLY ha avuto 122 partite:
-----
80 partite, 38 vittorie e 47.5% di prestazione -- Dentro casa
42 partite, 14 vittorie e 33.33% di prestazione -- Fuori casa

80 partite, 26 pareggi e 32.5% di prestazione -- Dentro casa
42 partite, 11 pareggi e 26.19% di prestazione -- Fuori casa

80 partite, 16 sconfitte e 20.0% di prestazione -- Dentro casa
42 partite, 17 sconfitte e 40.48% di prestazione -- Fuori casa
Italy nel torneo UEFAEUROQUALIFICATION ha avuto 66 partite:
-----
33 partite, 27 vittorie e 81.82% di prestazione -- Dentro casa

```

```

33 partite, 23 vittorie e 69.7% di prestazione -- Fuori casa

33 partite, 4 pareggi e 12.12% di prestazione -- Dentro casa
33 partite, 8 pareggi e 24.24% di prestazione -- Fuori casa

33 partite, 2 sconfitte e 6.06% di prestazione -- Dentro casa
33 partite, 2 sconfitte e 6.06% di prestazione -- Fuori casa
Italy nel torneo FIFAWORLDCUPQUALIFICATION ha avuto 72 partite:
-----
37 partite, 28 vittorie e 75.68% di prestazione -- Dentro casa
35 partite, 18 vittorie e 51.43% di prestazione -- Fuori casa

37 partite, 8 pareggi e 21.62% di prestazione -- Dentro casa
35 partite, 14 pareggi e 40.0% di prestazione -- Fuori casa

37 partite, 1 sconfitte e 2.7% di prestazione -- Dentro casa
35 partite, 3 sconfitte e 8.57% di prestazione -- Fuori casa
Italy nel torneo UEFAEURO ha avuto 34 partite:
-----
17 partite, 11 vittorie e 64.71% di prestazione -- Dentro casa
17 partite, 6 vittorie e 35.29% di prestazione -- Fuori casa

17 partite, 5 pareggi e 29.41% di prestazione -- Dentro casa
17 partite, 7 pareggi e 41.18% di prestazione -- Fuori casa

17 partite, 1 sconfitte e 5.88% di prestazione -- Dentro casa
17 partite, 4 sconfitte e 23.53% di prestazione -- Fuori casa
Italy nel torneo COPAAMÉRICA ha avuto 0 partite:
-----

ATTENZIONE!
Mi sa che Italy non ha mai partecipato al torneo: CopaAmérica.

```

▼ 4.3. Media di Gol per torneo

```

torneo_europa = ['FIFAWorldCup', 'Friendly', 'UEFAEuroqualification', 'FIFAWorldCupqualification', 'UEFAEuro', 'CopaAmérica']

def media_goal(torneo:str, vedere: int):
    torneo_df = df[df['tournament'] == torneo]
    totale, _ = torneo_df.shape
    goal_1 = torneo_df[['home_team_score']].sum()
    goal_2 = torneo_df[['away_team_score']].sum()

    goal_1 = goal_1.values.tolist()
    goal_2 = goal_2.values.tolist()
    #print(goal_1[0], goal_2[0])
    #print(goal_1[0] + goal_2[0])
    #print(totale)

    media = (goal_1[0] + goal_2[0]) / totale

    return round(media, 2)

for i in torneo_europa:
    print(f'In {i}: media di gol {media_goal(i, 0)}')

In FIFAWorldCup: media di gol 2.54
In Friendly: media di gol 2.51
In UEFAEuroqualification: media di gol 2.79
In FIFAWorldCupqualification: media di gol 2.87
In UEFAEuro: media di gol 2.4
In CopaAmérica: media di gol 2.6

```

▼ 4.4. FIFA Rank - Come si è comportata la nazionale negli ultimi 30 anni

```

paese_da_cercare = 'Italy'
rank = df[['date', 'home_team', 'away_team', 'home_team_fifa_rank', 'away_team_fifa_rank']]
rank = rank[(rank['home_team'] == paese_da_cercare) | (rank['away_team'] == paese_da_cercare)]
rank.head(5)

```

	date	home_team	away_team	home_team_fifa_rank	away_team_fifa_rank
54	1993-09-22	Estonia	Italy	118	2
83	1993-10-13	Italy	Scotland	2	28
133	1993-11-17	Italy	Portugal	3	20
220	1994-02-16	Italy	France	7	14
251	1994-03-23	Germany	Italy	1	11

```
home_team_fifa_rank = rank[['date', 'home_team', 'home_team_fifa_rank']]
home_team_fifa_rank = home_team_fifa_rank[home_team_fifa_rank['home_team'] == paese_da_cercare]
home_team_lista = home_team_fifa_rank.values.tolist()
print(home_team_lista)
home_team_fifa_rank.head()
```

```
[['1993-10-13', 'Italy', 2], ['1993-11-17', 'Italy', 3], ['1994-02-16', 'Italy', 7], ['1994-05-27', 'Italy', 16], ['1994-06-03', 'It
```

	date	home_team	home_team_fifa_rank
83	1993-10-13	Italy	2
133	1993-11-17	Italy	3
220	1994-02-16	Italy	7
342	1994-05-27	Italy	16
360	1994-06-03	Italy	16

```
away_team_fifa_rank = rank[['date', 'away_team', 'away_team_fifa_rank']]
away_team_fifa_rank = away_team_fifa_rank[away_team_fifa_rank['away_team'] == paese_da_cercare]
away_team_lista = away_team_fifa_rank.values.tolist()
print(away_team_lista)
away_team_fifa_rank.head()
```

```
[('1993-09-22', 'Italy', 2), ('1994-03-23', 'Italy', 11), ('1994-06-11', 'Italy', 16), ('1994-07-05', 'Italy', 4), ('1994-07-13', 'I
```

	date	away_team	away_team_fifa_rank
54	1993-09-22	Italy	2
251	1994-03-23	Italy	11
377	1994-06-11	Italy	16
430	1994-07-05	Italy	4
439	1994-07-13	Italy	4

```
lista = home_team_lista + away_team_lista
print(lista)
```

```
[('1993-10-13', 'Italy', 2), ('1993-11-17', 'Italy', 3), ('1994-02-16', 'Italy', 7), ('1994-05-27', 'Italy', 16), ('1994-06-03', 'It
```

```
anni = []
```

```
for x in lista:
    anno = x[0]
    anno = anno.split(sep='-')[0]
    anni.append(anno)
```

```
anni = list(set(anni))
anni.sort()
print(anni)
```

['1993', '1994', '1995', '1996', '1997', '1998', '1999', '2000', '2001', '2002', '2003', '2004', '2005', '2006', '2007', '2008', '20

```
def media_punteggi_degli_anni(tutto: list, gli_anni: list):
    media_degli_anni = []
    for anno in gli_anni:
        punteggi = []
        for x in tutto:
            a = x[0]
            a = a.split(sep='-')[0]
            p = x[2]
            if a == anno:
                punteggi.append(p)
        media_punteggi = round((reduce(lambda x, y: x + y, punteggi)) / len(punteggi), 2)
        media_degli_anni.append(media_punteggi)
    return media_degli_anni
```

```
medie = media_punteggi_degli_anni(lista, anni)
print(medie)
```

[2.33, 6.5, 3.56, 6.5, 9.75, 12.0, 5.11, 11.64, 4.22, 7.08, 11.7, 9.69, 11.58, 9.19, 2.1, 2.69, 4.07, 8.5, 8.25, 9.36, 6.67, 10.75,


```
def graficos(liata_x:list, lista_y:list, colere_x:str, colore_y:str, nome:str, titolo:str):
```

```
import matplotlib.pyplot as plt
x = liata_x
y = lista_y

plt.plot(x,y,color=colere_x)
plt.plot(x,y,color=colore_y)
plt.xticks(rotation = 90)
plt.title(titolo)
plt.savefig(nome + '.jpg')
plt.show()
```

```
graficos(anni, medie, 'blue', 'black', 'Italia', 'Prestazioni')
```



▼ 4.5. FIFA Rank - La classifica

```
# ['SolomonIslands', 'Panama', 'Guinea', 'Thailand', 'Chad', 'Bahamas', 'Philippines', 'Guam', 'St.KittsandNevis', 'Kazakhstan', 'USA', '']
```

```
migliori1 = df[['home_team', 'home_team_fifa_rank']]
migliori1 = migliori1[((migliori1['home_team_fifa_rank'] > 0) & (migliori1['home_team_fifa_rank'] < 16))]
migliori1 = migliori1.values.tolist()
print(len(migliori1))
```

```
2836
```

```
migliori2 = df[['away_team', 'away_team_fifa_rank']]
migliori2 = migliori2[((migliori2['away_team_fifa_rank'] > 0) & (migliori2['away_team_fifa_rank'] < 16))]
migliori2 = migliori2.values.tolist()
print(len(migliori2))
```

```
2515
```

```
migliori = migliori1 + migliori2
print(migliori)
```

```
nazionali = []
```

```
for x in migliori:
    nazionali.append(x[0])
```

```
nazionali = list(set(nazionali))
print(nazionali)
```

```
[['Brazil', 8], ['Sweden', 4], ['Argentina', 5], ['Brazil', 8], ['Sweden', 4], ['Denmark', 10], ['Argentina', 5], ['Brazil', 8], ['Switzerland', 'Slovenia', 'Algeria', 'Chile', 'Ecuador', 'Croatia', 'Uruguay', 'Nigeria', 'Mexico', 'Ukraine', 'France', 'Peru',
```

```
def fifa_rank(paese):
    # Prendendo le partite giocate dalla squadra
    rank = df[['date', 'home_team', 'away_team', 'home_team_fifa_rank', 'away_team_fifa_rank']]
    rank = rank[(rank['home_team'] == paese) | (rank['away_team'] == paese)]
```

```

# Giocate dentro casa
home_team_fifa_rank = rank[['date', 'home_team', 'home_team_fifa_rank']] # home_team_fifa_rank # home_team_score
home_team_fifa_rank = home_team_fifa_rank[home_team_fifa_rank['home_team'] == paese]
home_team_lista = home_team_fifa_rank.values.tolist()

# Giocate fuori casa
away_team_fifa_rank = rank[['date', 'away_team', 'away_team_fifa_rank']] # away_team_fifa_rank # away_team_score
away_team_fifa_rank = away_team_fifa_rank[away_team_fifa_rank['away_team'] == paese]
away_team_lista = away_team_fifa_rank.values.tolist()

# Lista principale (la somma delle partite dentro e fuori casa)
lista_principale = home_team_lista + away_team_lista

return lista_principale

paese_da_cercare = 'Italy'

lista = fifa_rank(paese_da_cercare)
print(lista)

[['1993-10-13', 'Italy', 2], ['1993-11-17', 'Italy', 3], ['1994-02-16', 'Italy', 7], ['1994-05-27', 'Italy', 16], ['1994-06-03', 'It

<

```

```

anni = []

for x in lista:
    anno = x[0]
    anno = anno.split(sep='-')[0]
    anni.append(anno)

anni = list(set(anni))
anni.sort()
print(anni)

['1993', '1994', '1995', '1996', '1997', '1998', '1999', '2000', '2001', '2002', '2003', '2004', '2005', '2006', '2007', '2008', '20

<

```

```

def media_punteggi_degli_anni(tutto: list, gli_anni: list):
    media_degli_anni = []
    for anno in gli_anni:
        punteggi = []
        for x in tutto:
            a = x[0]
            a = a.split(sep='-')[0]
            p = x[2]
            if a == anno:
                punteggi.append(p)
        media_punteggi = round((reduce(lambda x, y: x + y, punteggi)) / len(punteggi), 2)
        media_degli_anni.append(media_punteggi)
    return media_degli_anni

medie = media_punteggi_degli_anni(lista, anni)
media_del_periodo = round(reduce(lambda x, y: x + y, medie) / len(medie), 2)
string = paese_da_cercare, media_del_periodo
classifica = []
classifica.append(string)

print(f">>> {paese_da_cercare} ha preso {media_del_periodo} come punteggio medio nell'ultimo periodo.\n")
print(classifica)

>>> Italy ha preso 8.78 come punteggio medio nell'ultimo periodo.

[('Italy', 8.78)]

def calcola_classifica(anni, nazionali, periodo:None):
    classifica = []
    for nazione in nazionali:
        lista_del_paese = fifa_rank(nazione)
        if periodo != None:
            periodo_lista = []
            for x in lista_del_paese:
                a = x[0]
                a = a.split(sep='-')[0]
                if a == periodo:
                    periodo_lista.append(x)
            lista_del_paese = periodo_lista
            anni = [periodo]
    try:

```

```

        medie_del_paese = media_punteggi_degli_anni(lista_del_paese, anni)
    except TypeError:
        continue
    media_del_paese_nel_periodo = round(reduce(lambda x, y: x + y, medie_del_paese) / len(medie_del_paese), 2)
    string = [nazionele, media_del_paese_nel_periodo]
    classifica.append(string)
return classifica

classifica = calcola_classifica(anni, nazionali, None)

def lista_in_ordine(lista:list, indice:int, smistaggio:None) -> list:
    lista_ordinata = sorted(lista, key=lambda x: x[indice])
    if smistaggio != None:
        ordine = 0
        for x in lista_ordinata:
            ordine += 1
            print(f'{ordine}° {x[0]} con il {x[indice]}')
    return lista_ordinata

l = lista_in_ordine(classifica, 1, 1)
print(l)

1° Brazil con il 3.19
2° Spain con il 5.93
3° Argentina con il 6.34
4° Germany con il 6.47
5° France con il 8.37
6° Italy con il 8.78
7° England con il 10.07
8° Netherlands con il 10.19
9° Portugal con il 11.38
10° Mexico con il 14.42
11° Denmark con il 18.76
12° Colombia con il 19.32
13° USA con il 20.29
14° Sweden con il 21.02
15° Uruguay con il 22.48
16° Romania con il 25.45
17° Belgium con il 26.22
18° Switzerland con il 28.45
19° Chile con il 30.84
20° Turkey con il 31.82
21° Norway con il 32.05
22° Greece con il 32.72
23° RepublicofIreland con il 32.98
24° Poland con il 34.84
25° Japan con il 34.84
26° Tunisia con il 35.14
27° Cameroon con il 38.83
28° Côte d'Ivoire con il 39.29
29° Nigeria con il 39.35
30° Scotland con il 41.19
31° Ukraine con il 42.39
32° Ecuador con il 43.37
33° Morocco con il 43.52
34° Ghana con il 43.52
35° Costa Rica con il 44.96
36° Bulgaria con il 44.98
37° Austria con il 47.17
38° Algeria con il 52.85
39° Israel con il 54.47
40° Wales con il 56.07
41° Zambia con il 60.67
[['Brazil', 3.19], ['Spain', 5.93], ['Argentina', 6.34], ['Germany', 6.47], ['France', 8.37], ['Italy', 8.78], ['England', 10.07],

```

▼ 5. I grafici

```

def graficos(liata_x:list, lista_y:list, colere_x:str, colore_y:str, nome:str, titolo:str):

    import matplotlib.pyplot as plt
    x = liata_x
    y = lista_y

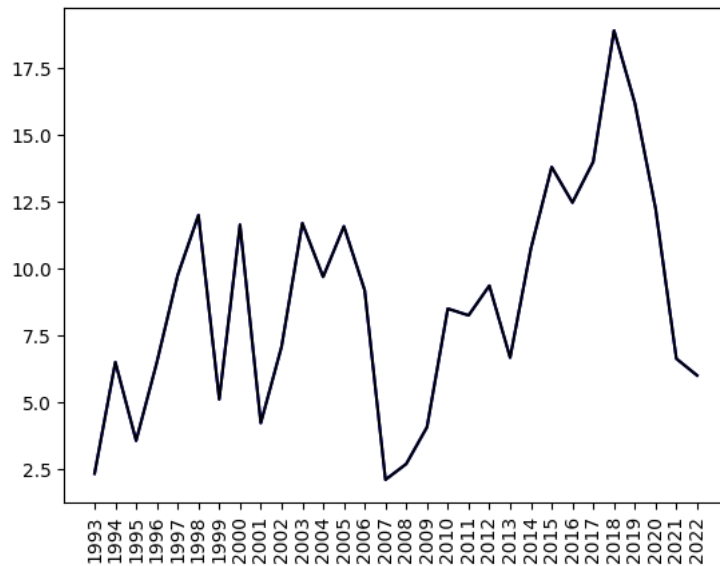
    plt.plot(x,y,color=colere_x)
    plt.plot(x,y,color=colore_y)
    plt.xticks(rotation = 90)
    plt.title(titolo)
    plt.savefig(nome + '.jpg')
    plt.show()

graficos(anni, medie, 'blue', 'black', 'Italia', 'Prestazioni')

```



Prestazioni



6. Storytelling

FIFA Rank - Le prime 15

```
import random
```

```
# Crea una lista di 30 numeri casuali tra 1 e 100
```

```
#numeri_casuali = [random.randint(1, 100) for _ in range(30)]
```

```
numeri_casuali = [58, 79, 82, 49, 10, 23, 32, 92, 41, 78, 80, 21, 41, 11, 79, 39, 80, 96, 91, 12, 27, 32, 78, 74, 44, 59, 72, 64, 12, 65]  
print(numeri_casuali) # Stampa la lista di numeri casuali
```

```
[58, 79, 82, 49, 10, 23, 32, 92, 41, 78, 80, 21, 41, 11, 79, 39, 80, 96, 91, 12, 27, 32, 78, 74, 44, 59, 72, 64, 12, 65]
```

```
paese_da_cercare = 'Italy'
```

```
rank = df[['home_team', 'home_team_fifa_rank']]
```

```
home_team_fifa_rank = rank[((rank['home_team_fifa_rank'] > 0) & (rank['home_team_fifa_rank'] < 16))]  
home_team_fifa_rank.head()
```

	home_team	home_team_fifa_rank
1	Brazil	8
9	Sweden	4
15	Argentina	5
17	Brazil	8
19	Sweden	4

```
rank = df[['away_team', 'away_team_fifa_rank']]
```

```
away_team_fifa_rank = rank[((rank['away_team_fifa_rank'] > 0) & (rank['away_team_fifa_rank'] < 16))]  
away_team_fifa_rank.head()
```

	away_team	away_team_fifa_rank
1	Mexico	14
4	Argentina	5
8	Norway	9
9	Switzerland	3
12	Argentina	5

[illegible]

	home_team	home_team_fifa_rank	away_team	away_team_fifa_rank
220	Italy	7	France	14
251	Germany	1	Italy	11
388	Italy	4	RepublicofIreland	14
400	Italy	4	Norway	6
430	Nigeria	11	Italy	4