

## Importation des bibliotheques

```
In [ ]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
matches = pd.read_csv("results.csv")
rankings = pd.read_csv("fifa_ranking-2022-10-06.csv")
```

## preparation des donnees

```
In [3]: rankings["rank_date"] = pd.to_datetime(rankings["rank_date"])
matches = matches.replace({"Germany DR": "Germany", "China": "China PR"})
rankings = rankings.replace({"IR Iran": "Iran"})
matches["date"] = pd.to_datetime(matches["date"])
```

```
In [4]: matches.head()
```

```
Out[4]:
```

	date	home_team	away_team	home_score	away_score	tournament	city	country	neut
0	1872-11-30	Scotland	England	0	0	Friendly	Glasgow	Scotland	Fa
1	1873-03-08	England	Scotland	4	2	Friendly	London	England	Fa
2	1874-03-07	Scotland	England	2	1	Friendly	Glasgow	Scotland	Fa
3	1875-03-06	England	Scotland	2	2	Friendly	London	England	Fa
4	1876-03-04	Scotland	England	3	0	Friendly	Glasgow	Scotland	Fa

## creation d'une nouveau dataframe

on a calculer la moyenne des buts marqués et encaissés par chaque équipe nationale et injecter dans une nouveau dataframe qu'on l'appelle (df\_team\_strength).

```
In [26]: result=matches[(matches['tournament']=='FIFA World Cup')]
```

```
In [29]: home_team = result[['home_team', 'home_score', 'away_score']].rename(
        columns={'home_team': 'Team', 'home_score': 'Scored', 'away_score': 'Conceded'})

away_team = result[['away_team', 'home_score', 'away_score']].rename(
        columns={'away_team': 'Team', 'away_score': 'GoalsScored', 'home_score': 'GoalsCor

Teams_result = pd.concat([home_team,away_team])
df_team_strength=Teams_result.groupby(['Team']).mean()
```

```
In [28]: Teams_result
```

Out[28]:

	Scored	Conceded
Team		
Algeria	1.000000	1.461538
Angola	0.333333	0.666667
Argentina	1.694118	1.129412
Australia	0.850000	1.850000
Austria	1.482759	1.620690
...	...	...
United Arab Emirates	0.666667	3.666667
United States	1.081081	1.783784
Uruguay	1.508475	1.288136
Wales	0.625000	1.250000
Yugoslavia	1.666667	1.272727

82 rows × 2 columns

## Construction du modèle

j'ai décidé de prendre la distribution de Poisson comme un modèle de prédiction, La distribution de Poisson est une distribution de probabilité discrète qui décrit le nombre d'événements se produisant dans un intervalle de temps fixe ou une région d'opportunité. Si nous considérons un but comme un événement susceptible de se produire au cours des 90 minutes d'un match de football, nous pourrions calculer la probabilité du nombre de buts qui pourraient être marqués au cours d'un match par l'équipe A et l'équipe B.

```
In [30]: def predict_points(home, away):
    if home in df_team_strength.index and away in df_team_strength.index:
        lamb_home = df_team_strength.at[home, 'GoalsScored'] * df_team_strength.at[home, 'GoalsConceded']
        lamb_away = df_team_strength.at[away, 'GoalsScored'] * df_team_strength.at[away, 'GoalsConceded']
        prob_home, prob_away, prob_draw = 0, 0, 0
        for x in range(0, 11): # nombre de buts de l'équipe à domicile
            for y in range(0, 11): # nombre de buts de l'équipe en déplacement
                p = poisson.pmf(x, lamb_home) * poisson.pmf(y, lamb_away)
                if x == y:
                    prob_draw += p
                elif x > y:
                    prob_home += p
                else:
                    prob_away += p

        points_home = 3 * prob_home + prob_draw
        points_away = 3 * prob_away + prob_draw
        return (points_home, points_away)
    else:
        return (0, 0)
```

## Application du modèle

```
In [41]: from scipy.stats import poisson
predict_points('England', 'United States')
```

```
Out[41]: (2.063001339164194, 0.7513810968805393)
```

```
In [42]: def get_winner(df_fixture_updated):
    for index, row in df_fixture_updated.iterrows():
        home, away = row['home'], row['away']
        points_home, points_away = predict_points(home, away)
        if points_home > points_away:
            winner = home
        else:
            winner = away
        df_fixture_updated.loc[index, 'winner'] = winner
    return df_fixture_updated
```

```
In [43]: df4 = pd.DataFrame ({ 'home': ['Morocco', "Croatia"] , 'away': ['France' ,"Argentina"] })
```

```
In [44]: get_winner(df4)
```

```
Out[44]:
```

	home	away	winner
0	Morocco	France	Morocco
1	Croatia	Argentina	Croatia

```
In [45]: df5 = pd.DataFrame ({ 'home': ['Morocco'] , 'away': ['Croitia']} )
```

```
In [46]: get_winner(df5)
```

```
Out[46]:
```

	home	away	winner
0	Morocco	Croitia	Croitia

```
In [ ]:
```