

```
# Rafael Ceotto  
# 12/3/2023  
# Project Research - Brazilian Soccer Championship  
Prediction
```

```
# Data from Wikipedia:
```

```
Link:https://pt.wikipedia.org/wiki/Campeonato\_Brasileiro\_de\_Futebol\_de\_2023\_-\_S%C3%A9rie\_A
```

```
# Using the distribution of Poisson,  
# In a nutshell, Poisson distributon is a statistic distribution  
that calculates the probability of a result to happen and all of  
the next results as well.
```

```
Link:https://www.itl.nist.gov/div898/handbook/eda/section3/eda366j.htm
```

```
In [1]: import pandas as pd
import requests
#Scipy library already been added to anaconda - taking Poisson distribution from that
from scipy.stats import poisson
import numpy as np

data_request = requests.get("https://pt.wikipedia.org/wiki/Campeonato_Brasileiro_de_Futebol_de_20
origin_table = pd.read_html(data_request.text)

classification_table = origin_table[6]
games_table = origin_table[7]
display(classification_table)
display(games_table)
```

	Pos	Equipe	pts	J	V	E	D	GP	GC	SG	Classificação ou descenso
0	1	Palmeiras (Q)	69	37	20	9	8	63	32	+31	Fase de grupos da Copa Libertadores de 2024
1	2	Atlético Mineiro (T)	66	37	19	9	9	51	28	+23	Fase de grupos da Copa Libertadores de 2024
2	3	Flamengo (T)	66	37	19	9	9	56	41	+15	Fase de grupos da Copa Libertadores de 2024
3	4	Grêmio (T)	65	37	20	5	12	60	54	+6	Fase de grupos da Copa Libertadores de 2024
4	5	Botafogo (T)	64	37	18	10	9	57	34	+23	Segunda fase da Copa Libertadores de 2024
5	6	Red Bull Bragantino (Q)	62	37	17	11	9	48	33	+15	Segunda fase da Copa Libertadores de 2024
6	7	Fluminense (Q)	56	37	16	8	13	49	44	+5	Fase de grupos da Copa Libertadores de 2024[a]
7	8	Athletico Paranaense (Q)	56	37	14	14	9	51	40	+11	Fase de grupos da Copa Sul-Americana de 2024
8	9	Internacional (Q)	52	37	14	10	13	43	44	-1	Fase de grupos da Copa Sul-Americana de 2024
9	10	Fortaleza (Q)	51	37	14	9	14	43	43	0	Fase de grupos da Copa Sul-Americana de 2024
10	11	São Paulo (Q)	50	37	13	11	13	39	38	+1	Fase de grupos da Copa Libertadores de 2024[b]
11	12	Cuiabá (Q)	48	37	13	9	15	37	39	-2	Fase de grupos da Copa Sul-Americana de 2024
12	13	Corinthians (Q)	47	37	11	14	12	45	48	-3	Fase de grupos da Copa Sul-Americana de 2024
13	14	Cruzeiro	46	37	11	13	13	34	31	+3	Fase de grupos da Copa Sul-Americana de 2024
14	15	Santos	43	37	11	10	16	38	62	-24	NaN
15	16	Vasco da Gama	42	37	11	9	17	39	50	-11	NaN
16	17	Bahia	41	37	11	8	18	46	52	-6	Rebaixados à Série B de 2024
17	18	Goiás (R)	35	37	8	11	18	35	53	-18	Rebaixados à Série B de 2024
18	19	Coritiba (R)	30	37	8	6	23	41	71	-30	Rebaixados à Série B de 2024
19	20	América Mineiro (R)	24	37	5	9	23	42	80	-38	Rebaixados à Série B de 2024

	Casa \ Fora	AMM	ATP	ATM	BAH	BOT	COR	CTB	CRU	CUI	...	FLU	FOR	GOI	GRE	INT	PAL	RBB	SAN
0	América Mineiro	—	2-2	1-1	3-2	1-2	2-0	0-3	0-4	1-2	...	0-3	2-1	0-1	3-4	1- ₂	1-4	0-2	2-0
1	Athletico Paranaense	3-2	—	1-1	2-0	1-0	1-0	3-2	3-3	2-0	...	2-2	1-1	2-0	1-2	2- ₁	2-2	1-1	3-0
2	Atlético Mineiro	2-2	2-1	—	1-0	1-0	0-1	1-2	0-1	1-0	...	2-0	3-1	2-1	3-0	2- ₀	1-1	1-1	2-0
3	Bahia	3-1	1-1	NaN	—	1-2	0-0	3-1	2-2	0-3	...	1-0	2-0	1-1	1-2	1- ₀	1-0	4-0	1-2
4	Botafogo	2-0	1-1	2-0	3-0	—	3-0	4-1	0-0	0-1	...	1-0	2-0	1-1	3-4	3- ₁	3-4	2-0	1-1
5	Corinthians	1-1	1-0	1-1	1-5	1-0	—	3-1	2-1	1-1	...	2-0	1-1	1-1	4-4	1- ₂	0-0	0-1	1-1
6	Coritiba	3-1	2-0	1-2	2-4	1-1	NaN	—	1-0	0-3	...	2-0	0-3	0-1	1-2	0- ₁	0-2	0-1	0-0
7	Cruzeiro	1-1	1-1	0-1	3-0	0-0	1-1	0-0	—	0-1	...	0-2	0-1	0-1	1-0	1- ₂	a	0-0	2-1
8	Cuiabá	2-2	NaN	0-4	1-1	0-1	0-1	1-1	0-0	—	...	3-0	2-1	1-1	1-2	0- ₂	0-2	1-1	3-0
9	Flamengo	1-1	0-3	0-3	1-0	2-3	1-0	3-0	1-1	2-1	...	1-1	2-0	2-0	3-0	0- ₀	3-0	1-0	1-2
10	Fluminense	3-1	2-0	1-1	2-1	0-2	3-3	2-1	1-0	2-0	...	—	1-0	5-3	a	2- ₀	2-1	2-1	1-0
11	Fortaleza	3-2	1-0	2-1	0-0	2-2	2-1	3-1	0-1	0-1	...	4-2	—	1-0	1-1	1- ₁	2-2	0-3	4-0
12	Goiás	NaN	1-1	0-0	4-6	2-1	3-1	1-2	0-1	0-1	...	2-2	1-0	—	1-1	0- ₀	0-5	0-2	0-1
13	Grêmio	3-1	1-2	1-0	1-0	0-2	0-1	5-1	3-0	2-0	...	2-1	0-0	2-1	—	3- ₁	1-0	3-3	1-0
14	Internacional	1-1	0-2	0-2	2-0	NaN	2-2	3-4	0-0	1-2	...	0-0	0-1	1-0	3-2	—	0-0	1-0	7-1
15	Palmeiras	4-0	1-0	0-2	1-0	0-1	2-1	3-1	1-0	2-1	...	1-0	3-1	1-0	4-1	3- ₀	—	1-1	1-2
16	Red Bull Bragantino	2-2	2-0	1-2	2-1	2-2	1-0	1-0	0-3	2-0	...	1-0	1-2	2-0	2-0	0- ₀	2-1	—	2-0
17	Santos	3-2	1-1	0-0	3-0	2-2	0-2	2-1	0-3	0-0	...	0-3	NaN	4-3	2-1	1- ₁	0-0	1-3	—
18	São Paulo	3-0	2-1	0-2	0-0	0-0	2-1	2-1	1-0	0-0	...	1-0	1-2	2-1	3-0	2- ₀	0-2	1-0	4-1
19	Vasco da Gama	2-1	0-2	1-0	0-1	1-0	2-4	5-1	0-1	1-0	...	4-2	1-0	0-1	1-0	1- ₂	2-2	NaN	0-1

20 rows × 21 columns



```
In [2]: team_names = list(games_table["Casa \ Fora"])
nicknames = list(games_table.columns)
nicknames.pop(0)
name_from_to = dict(zip(nicknames, team_names))
print(name_from_to)
```

```
{'AMM': 'América Mineiro', 'ATP': 'Athletico Paranaense', 'ATM': 'Atlético Mineiro', 'BAH': 'Bahia', 'BOT': 'Botafogo', 'COR': 'Corinthians', 'CTB': 'Coritiba', 'CRU': 'Cruzeiro', 'CUI': 'Cuiabá', 'FLA': 'Flamengo', 'FLU': 'Fluminense', 'FOR': 'Fortaleza', 'GOI': 'Goiás', 'GRE': 'Grêmio', 'INT': 'Internacional', 'PAL': 'Palmeiras', 'RBB': 'Red Bull Bragantino', 'SAN': 'Santos', 'SPA': 'São Paulo', 'VAS': 'Vasco da Gama'}
```

```
In [3]: updated_games_table = games_table.set_index("Casa \ Fora")
updated_games_table = updated_games_table.unstack().reset_index()
updated_games_table = updated_games_table.rename(columns={"level_0": "Out", "Casa \ Fora": "Home"})

def nickname_updated(line):
    nicknames = line["Out"]
    name = name_from_to[nicknames]
    return name

updated_games_table["Out"] = updated_games_table.apply(nickname_updated, axis=1)
updated_games_table = updated_games_table[updated_games_table["Out"]!=updated_games_table["Home"]]
display(updated_games_table)
```

	Out	Home	Results
1	América Mineiro	Athletico Paranaense	3-2
2	América Mineiro	Atlético Mineiro	2-2
3	América Mineiro	Bahia	3-1
4	América Mineiro	Botafogo	2-0
5	América Mineiro	Corinthians	1-1
...
394	Vasco da Gama	Internacional	2-1
395	Vasco da Gama	Palmeiras	1-0
396	Vasco da Gama	Red Bull Bragantino	1-1
397	Vasco da Gama	Santos	4-1
398	Vasco da Gama	São Paulo	4-2

380 rows × 3 columns

```
In [4]: missing_games = updated_games_table[~updated_games_table["Results"].str.contains("-",na=False)]
missing_games = updated_games_table.drop(columns=["Results"])
display(missing_games)
```

	Out	Home
1	América Mineiro	Athletico Paranaense
2	América Mineiro	Atlético Mineiro
3	América Mineiro	Bahia
4	América Mineiro	Botafogo
5	América Mineiro	Corinthians
...
394	Vasco da Gama	Internacional
395	Vasco da Gama	Palmeiras
396	Vasco da Gama	Red Bull Bragantino
397	Vasco da Gama	Santos
398	Vasco da Gama	São Paulo

380 rows × 2 columns

```
In [5]: played_games = updated_games_table[updated_games_table["Results"].str.contains("-", na=False)]
played_games[["Home Goals", "Out Goals"]] = played_games["Results"].str.split("-", expand=True)
played_games = played_games.drop(columns=["Results"])
played_games["Home Goals"] = played_games["Home Goals"].astype(int)
played_games["Out Goals"] = played_games["Out Goals"].astype(int)
display(played_games)
```

C:\Users\rafak\AppData\Local\Temp\ipykernel_11164\1478148269.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
played_games[["Home Goals", "Out Goals"]] = played_games["Results"].str.split("-", expand=True)
```

C:\Users\rafak\AppData\Local\Temp\ipykernel_11164\1478148269.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
played_games[["Home Goals", "Out Goals"]] = played_games["Results"].str.split("-", expand=True)
```

	Out	Home	Home Goals	Out Goals
1	América Mineiro	Athletico Paranaense	3	2
2	América Mineiro	Atlético Mineiro	2	2
3	América Mineiro	Bahia	3	1
4	América Mineiro	Botafogo	2	0
5	América Mineiro	Corinthians	1	1
...
394	Vasco da Gama	Internacional	2	1
395	Vasco da Gama	Palmeiras	1	0
396	Vasco da Gama	Red Bull Bragantino	1	1
397	Vasco da Gama	Santos	4	1
398	Vasco da Gama	São Paulo	4	2

370 rows × 4 columns

```
In [6]: home_goals_mean = played_games.groupby("Home").mean(numeric_only=True)
home_goals_mean = home_goals_mean.rename(columns={"Home Goals": "Goals at Home", "Out Goals": "Goals Against"})
display(home_goals_mean)
```

	Goals at Home	Goals Against
Home		
América Mineiro	1.105263	2.000000
Athletico Paranaense	1.736842	1.000000
Atlético Mineiro	1.473684	0.842105
Bahia	1.388889	1.111111
Botafogo	1.894737	0.894737
Corinthians	1.368421	1.210526
Coritiba	0.888889	1.388889
Cruzeiro	0.722222	0.888889
Cuiabá	1.111111	1.222222
Flamengo	1.368421	0.842105
Fluminense	1.722222	0.833333
Fortaleza	1.473684	1.052632
Goiás	1.000000	1.388889
Grêmio	1.789474	0.842105
Internacional	1.500000	1.111111
Palmeiras	1.842105	0.631579
Red Bull Bragantino	1.473684	0.736842
Santos	1.388889	1.444444
São Paulo	1.555556	0.722222
Vasco da Gama	1.222222	1.222222

```
In [7]: out_goals_mean = played_games.groupby("Out").mean(numeric_only=True)
out_goals_mean = out_goals_mean.rename(columns={"gols_casa": "Out Goals Against", "gols_fora": "Out Goals For"})
display(out_goals_mean)
```

	Home Goals	Out Goals
Out		
América Mineiro	2.333333	1.166667
Athletico Paranaense	1.166667	1.000000
Atlético Mineiro	0.666667	1.277778
Bahia	1.684211	1.105263
Botafogo	0.944444	1.166667
Corinthians	1.388889	1.055556
Coritiba	2.368421	1.263158
Cruzeiro	0.789474	1.105263
Cuiabá	0.894737	0.894737
Flamengo	1.388889	1.666667
Fluminense	1.526316	0.947368
Fortaleza	1.277778	0.833333
Goiás	1.473684	0.894737
Grêmio	2.111111	1.444444
Internacional	1.263158	0.842105
Palmeiras	1.111111	1.555556
Red Bull Bragantino	1.055556	1.111111
Santos	1.894737	0.684211
São Paulo	1.315789	0.578947
Vasco da Gama	1.421053	0.842105

```
In [8]: statistic_table = home_goals_mean.merge(out_goals_mean, left_index=True, right_index=True)
statistic_table = statistic_table.reset_index()
statistic_table = statistic_table.rename(columns={"Home": "Team"})
display(statistic_table)
```

	Team	Goals at Home	Goals Against	Home Goals	Out Goals
0	América Mineiro	1.105263	2.000000	2.333333	1.166667
1	Athletico Paranaense	1.736842	1.000000	1.166667	1.000000
2	Atlético Mineiro	1.473684	0.842105	0.666667	1.277778
3	Bahia	1.388889	1.111111	1.684211	1.105263
4	Botafogo	1.894737	0.894737	0.944444	1.166667
5	Corinthians	1.368421	1.210526	1.388889	1.055556
6	Coritiba	0.888889	1.388889	2.368421	1.263158
7	Cruzeiro	0.722222	0.888889	0.789474	1.105263
8	Cuiabá	1.111111	1.222222	0.894737	0.894737
9	Flamengo	1.368421	0.842105	1.388889	1.666667
10	Fluminense	1.722222	0.833333	1.526316	0.947368
11	Fortaleza	1.473684	1.052632	1.277778	0.833333
12	Goiás	1.000000	1.388889	1.473684	0.894737
13	Grêmio	1.789474	0.842105	2.111111	1.444444
14	Internacional	1.500000	1.111111	1.263158	0.842105
15	Palmeiras	1.842105	0.631579	1.111111	1.555556
16	Red Bull Bragantino	1.473684	0.736842	1.055556	1.111111
17	Santos	1.388889	1.444444	1.894737	0.684211
18	São Paulo	1.555556	0.722222	1.315789	0.578947
19	Vasco da Gama	1.222222	1.222222	1.421053	0.842105

1st Question - What's the probability for the São Paulo team to win against Coritiba?

R - According to the results, the house team(São Paulo) would have a better chance of winning, with 82.57%, against 8.3% of the game being a draw and 5.11% of the visitor team(Coritiba) winning.


```

In [9]: ▶ house = "São Paulo"
        visitor = "Coritiba"

def points_calculus(line):
    house = line["Home"]
    visitor = line["Out"]
    # Adding points to the current team
    estimated_value_winning = house_won_probability * 3 + draw
    estimated_value_losing = visitor_won_probability * 3 + draw
    line["In Home Points"] = estimated_value_winning
    line["Outside Points"] = estimated_value_losing
    return line

lambda_house = (statistic_table.loc[statistic_table["Team"]==house, "Goals at Home"].iloc[0]*
                statistic_table.loc[statistic_table["Team"]==visitor, "Home Goals"].iloc[0])

lambda_visitor = (statistic_table.loc[statistic_table["Team"]==visitor, "Out Goals"].iloc[0]
                  * statistic_table.loc[statistic_table["Team"]==house, "Goals Against"].iloc[0])

house_won_probability = 0
draw = 0
visitor_won_probability = 0

for set_home_goals in range(8):
    for set_out_goals in range(8):
        results_probability = poisson.pmf(set_home_goals, lambda_house) * poisson.pmf(set_out_goals)
        if set_home_goals == set_out_goals:
            draw += results_probability
        elif set_home_goals > set_out_goals:
            house_won_probability += results_probability
        elif set_home_goals < set_out_goals:
            visitor_won_probability += results_probability

pb_house = (house_won_probability * 100)
pb_draw = (draw * 100)
pb_visitor = (visitor_won_probability * 100)

print(round(np.mean(pb_house),2),'%')
print(round(np.mean(pb_draw),2),'%')
print(round(np.mean(pb_visitor),2),'%')

```

82.88 %
8.57 %
5.1 %

```
In [10]: ▶ missing_games = missing_games.apply(points_calculus, axis=1)
display(missing_games)
```

	Out	Home	In Home Points	Outside Points
1	América Mineiro	Athletico Paranaense	2.571974	0.238797
2	América Mineiro	Atlético Mineiro	2.571974	0.238797
3	América Mineiro	Bahia	2.571974	0.238797
4	América Mineiro	Botafogo	2.571974	0.238797
5	América Mineiro	Corinthians	2.571974	0.238797
...
394	Vasco da Gama	Internacional	2.571974	0.238797
395	Vasco da Gama	Palmeiras	2.571974	0.238797
396	Vasco da Gama	Red Bull Bragantino	2.571974	0.238797
397	Vasco da Gama	Santos	2.571974	0.238797
398	Vasco da Gama	São Paulo	2.571974	0.238797

380 rows × 4 columns

```
In [20]: def team_name_adjustment(line):
    for name in team_names:
        if name in line["Equipevde"]:
            return name

classification_table["Team"] = classification_table.apply(team_name_adjustment, axis=1)
classification_table_updated = classification_table[["Team", "Pts"]]
classification_table_updated["Pts"] = classification_table_updated["Pts"].astype(int)

display(classification_table_updated)
```

C:\Users\rafak\AppData\Local\Temp\ipykernel_11164\4240848832.py:8: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
classification_table_updated["Pts"] = classification_table_updated["Pts"].astype(int)
```

	Team	Pts
0	Palmeiras	69
1	Atlético Mineiro	66
2	Flamengo	66
3	Grêmio	65
4	Botafogo	64
5	Red Bull Bragantino	62
6	Fluminense	56
7	Athletico Paranaense	56
8	Internacional	52
9	Fortaleza	51
10	São Paulo	50
11	Cuiabá	48
12	Corinthians	47
13	Cruzeiro	46
14	Santos	43
15	Vasco da Gama	42
16	Bahia	41
17	Goiás	35
18	Coritiba	30
19	América Mineiro	24

2nd Question - After updating the games with current points, what team would be the champion and what team would be the last one?

R - After treating the data and updating to the latest prediction, Palmeiras would be the champion and America-MG would be demoted to the second division of Brazilian Soccer Championship

```
In [21]: ► classification_table["Team"] = classification_table.apply(team_name_adjustment, axis=1)
classification_table_updated = classification_table[["Team", "Pts"]]
classification_table_updated["Pts"] = classification_table_updated["Pts"].astype(int)
classification_table_updated.index = classification_table_updated.index + 1

display(classification_table_updated)
```

C:\Users\rafak\AppData\Local\Temp\ipykernel_11164\2143821265.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
classification_table_updated["Pts"] = classification_table_updated["Pts"].astype(int)
```

	Team	Pts
1	Palmeiras	69
2	Atlético Mineiro	66
3	Flamengo	66
4	Grêmio	65
5	Botafogo	64
6	Red Bull Bragantino	62
7	Fluminense	56
8	Athletico Paranaense	56
9	Internacional	52
10	Fortaleza	51
11	São Paulo	50
12	Cuiabá	48
13	Corinthians	47
14	Cruzeiro	46
15	Santos	43
16	Vasco da Gama	42
17	Bahia	41
18	Goiás	35
19	Coritiba	30
20	América Mineiro	24

In []: ►