

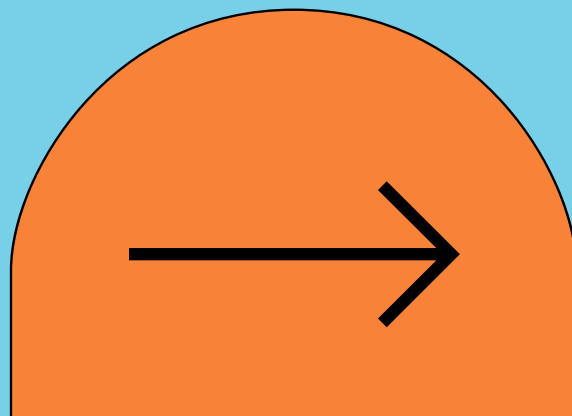
**Atividade  
17**



# Observação de pássaros

Equipe: Carlos Bahia, Jefferson Costa, Raquel Nunes.

# Sumário



1. Coleta de dados
2. Tratamento dos dados
3. Inserção dos dados no banco
4. Extração de dados do banco
5. Análise Gráfica
6. Insights

# 1. Coleta de dados:

**FONTE:**

<https://www.kaggle.com/>

**COLUNAS:**

Common.Name, Date, Time , n.observers, County, Sub.cell , Season,  
DEM, Cell.ID, List.ID

# Forma dos dados:

```
-----
```

	Common.Name	Date	Time	n.observers	County	Sub.cell	Season	DEM	Cell.ID	List.ID
0	Asian Koel	7/16/2015	16:30	2.0	Alappuzha	[51,2,2]	Wet	5.0	[76.28,9.84]	List.1
1	Black-rumped Flameback	7/16/2015	16:30	2.0	Alappuzha	[51,2,2]	Wet	5.0	[76.28,9.84]	List.1
2	Black Drongo	7/16/2015	16:30	2.0	Alappuzha	[51,2,2]	Wet	5.0	[76.28,9.84]	List.1
3	Brahminy Kite	7/16/2015	16:30	2.0	Alappuzha	[51,2,2]	Wet	5.0	[76.28,9.84]	List.1
4	Common Myna	7/16/2015	16:30	2.0	Alappuzha	[51,2,2]	Wet	5.0	[76.28,9.84]	List.1
...	...	...	...	...	...	...	...	...	...	...
300877	Great Hornbill	3/17/2019	10:30	2.0	Thrissur	NaN	Dry	616.0	NaN	NaN
300878	Great Hornbill	1/18/2019	9:05	8.0	Kollam	NaN	Dry	295.0	NaN	NaN
300879	Great Hornbill	2/8/2019	10:48	3.0	Palakkad	NaN	Dry	643.0	NaN	NaN
300880	Great Hornbill	1/26/2019	10:15	5.0	Idukki	NaN	Dry	795.0	NaN	NaN
300881	Great Hornbill	1/26/2019	10:00	5.0	Idukki	NaN	Dry	795.0	NaN	NaN

```
[300882 rows x 10 columns]
```

```
-----
```

# Códigos:

```
import pandas as pd
from sqlalchemy import create_engine
import matplotlib.pyplot as plt
import numpy as np
from modules.postgres import Conector_postgres
try:
    # ----- EXTRAÇÃO DOS DADOS -----
    df_dados = pd.read_csv("C:\\Users\\quel-\\OneDrive\\Área de Trabalho\\CURSOS\\Curso - Python\\Atividades_aulas_aovivo\\Ativid
    print(df_dados)
    print("-----")
```

## 2. Tratamento de dados



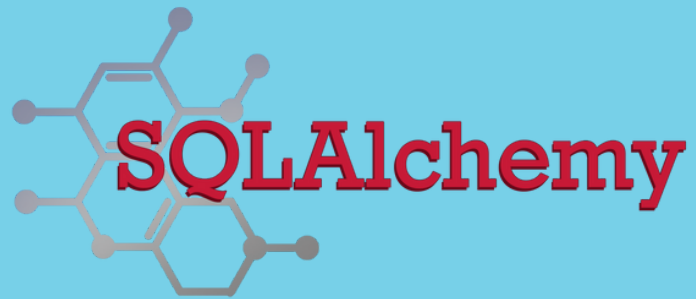
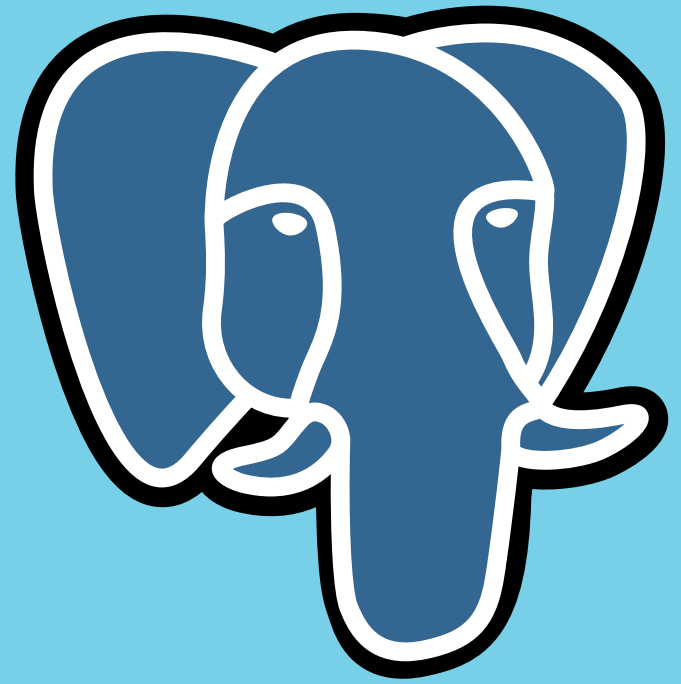
**Remoção de linhas duplicadas**

**Remoção de valores nulos**

**Remoção de colunas desnecessárias e renomeação das demais**

# Códigos:

```
# ----- TRATAMENTO DOS DADOS -----  
# Checando a presença de valores nulos:  
print("Quantidade de valores nulos por coluna: ")  
print(df_dados.isna().sum())  
print("-----")  
# Tratando os valores nulos  
df_dados.dropna(inplace=True)  
  
# Checando a presença de dados duplicados:  
print("Quantidade de linhas duplicadas df_dados: ", df_dados.duplicated().sum())  
  
# Tratando os dados duplicados:  
df_dados.drop_duplicates(inplace = True)  
  
# Removendo colunas desnecessárias:  
df_dados.drop(columns=['Sub.cell', 'Cell.ID'], inplace=True)  
  
# Trocando os nomes das colunas:  
df_dados = df_dados.rename(columns={"Common.Name" : 'common_name', "n.observers": 'n_observers', "Date": 'date_data', "Time": 'time_data',
```



### 3. Inserção dos dados no banco

① **Bibliotecas utilizadas:**

pandas  
psycopg2  
sqlalchemy

② **Comandos Utilizados:**

create\_engine  
to\_sql  
ALTER TABLE



# Códigos:

```
# ----- CARREGAMENTO DOS DADOS -----  
# Conectando com o banco de dados  
banco = Conector_postgres(host="127.0.0.1", db="atividade17")  
  
# Deletando as Views, caso existam  
banco.deletar("DROP VIEW IF EXISTS lista_data CASCADE")  
  
# Inserindo os valores no banco de dados:  
engine = create_engine('postgresql://postgres:postgres@127.0.0.1:5432/atividade17') #Insira aqui os seus dados, no seguinte formato:  
df_dados.to_sql('dados', engine, if_exists='replace', index=False)  
  
# Alterando os tipos de dados:  
banco.alterar("ALTER TABLE dados ALTER COLUMN date_data TYPE date USING date_data::date;"  
| | | "ALTER TABLE dados ALTER COLUMN time_data TYPE time USING time_data::time without time zone;")
```

# 4. Extração de dados do banco

## DESAFIO DA TABELA

Os dados utilizados não possuíam nenhuma coluna com valores únicos.

## CONSTRUÇÃO DE NOVOS DATA FRAMES

Através de comandos do tipo SELECT e VIEW foi possível realizar a criação de novos data frames agrupados para análise

# Códigos:

```
# ----- LENDO OS DADOS DO BANCO -----  
# Criando um data frame com a quantidade de espécies avistadas para cada um dos grupos, com o número de observadores no grupo e ordenado pelo  
df_observacoes = pd.DataFrame(banco.selecionar("SELECT list_id, COUNT(common_name) as num_especies, n_observers FROM dados GROUP BY (list_id,  
  
# Criando um data frame com a quantidade de espécies avistadas para cada grupo, ordenado por data e horário  
df_especies_grupos = pd.DataFrame(banco.selecionar("SELECT list_id, COUNT(common_name) as num_especies, date_data, time_data FROM dados GROUP  
  
# Criando um data frame com a quantidade de pássaros avistados por data, ordenado pela data  
df_passaros_dia = pd.DataFrame(banco.selecionar("SELECT date_data, COUNT(common_name) as num_especies FROM dados GROUP BY date_data ORDER BY  
df_passaros_dia.index = pd.to_datetime(df_passaros_dia['Data'])  
df_passaros_ano = df_passaros_dia.groupby(pd.Grouper(freq = 'Y')).sum()  
  
# Criando um data frame com a quantidade de visitas por dia, ordenado pela data  
banco.criar("CREATE VIEW lista_data AS (SELECT DISTINCT list_id, date_data FROM dados ORDER BY date_data);")  
df_visitas_dia = pd.DataFrame(banco.selecionar("SELECT date_data, COUNT(list_id) FROM lista_data GROUP BY date_data ORDER BY date_data;"), co  
df_visitas_dia.index = pd.to_datetime(df_visitas_dia['Data'])  
df_visitas_ano = df_visitas_dia.groupby(pd.Grouper(freq='Y')).sum()
```

# 5. Análise Gráfica

## Bibliotecas utilizadas

Pandas  
Matplotlib.pyplot  
numpy

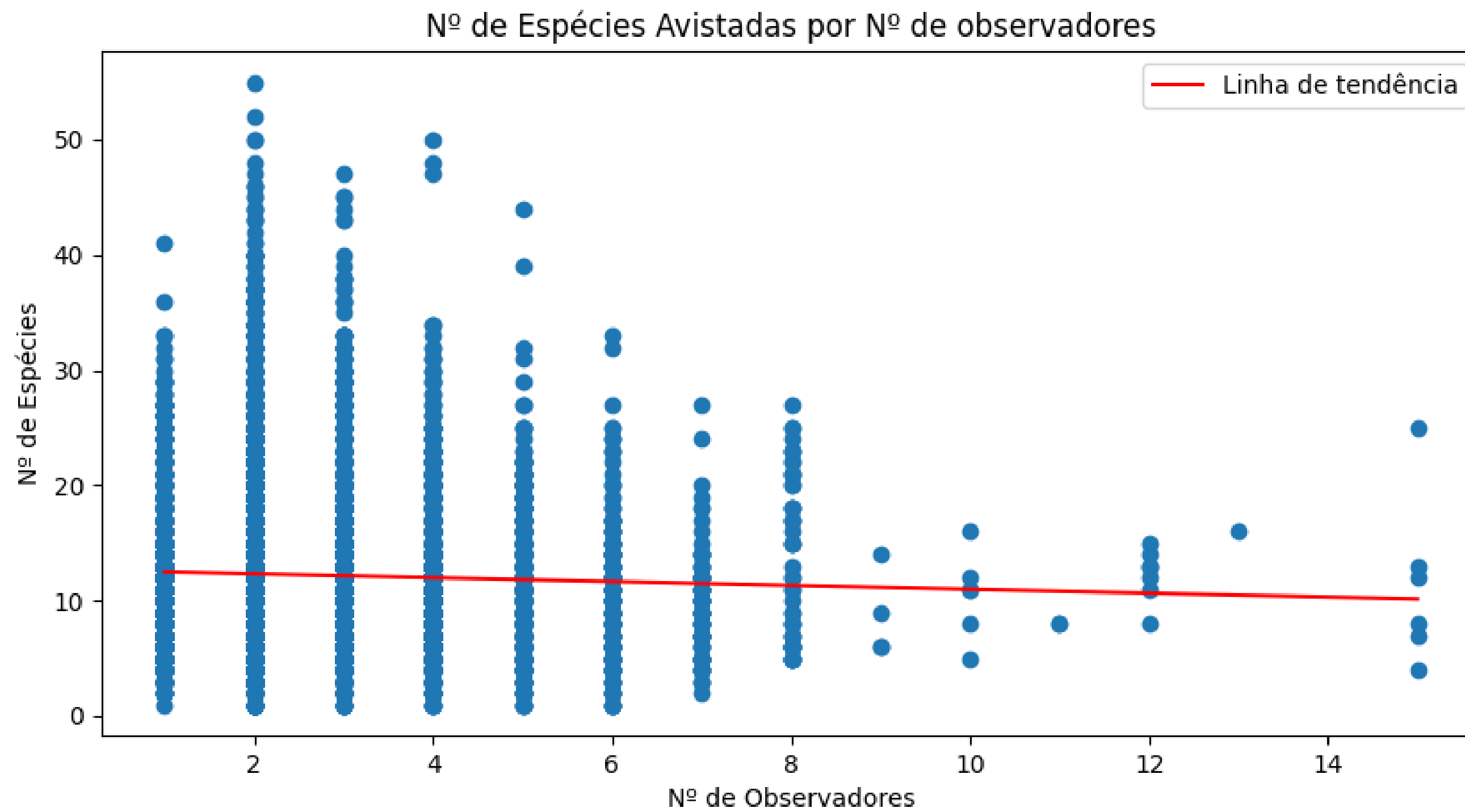
## Comandos

polyfit  
poly1d  
plot  
scatter

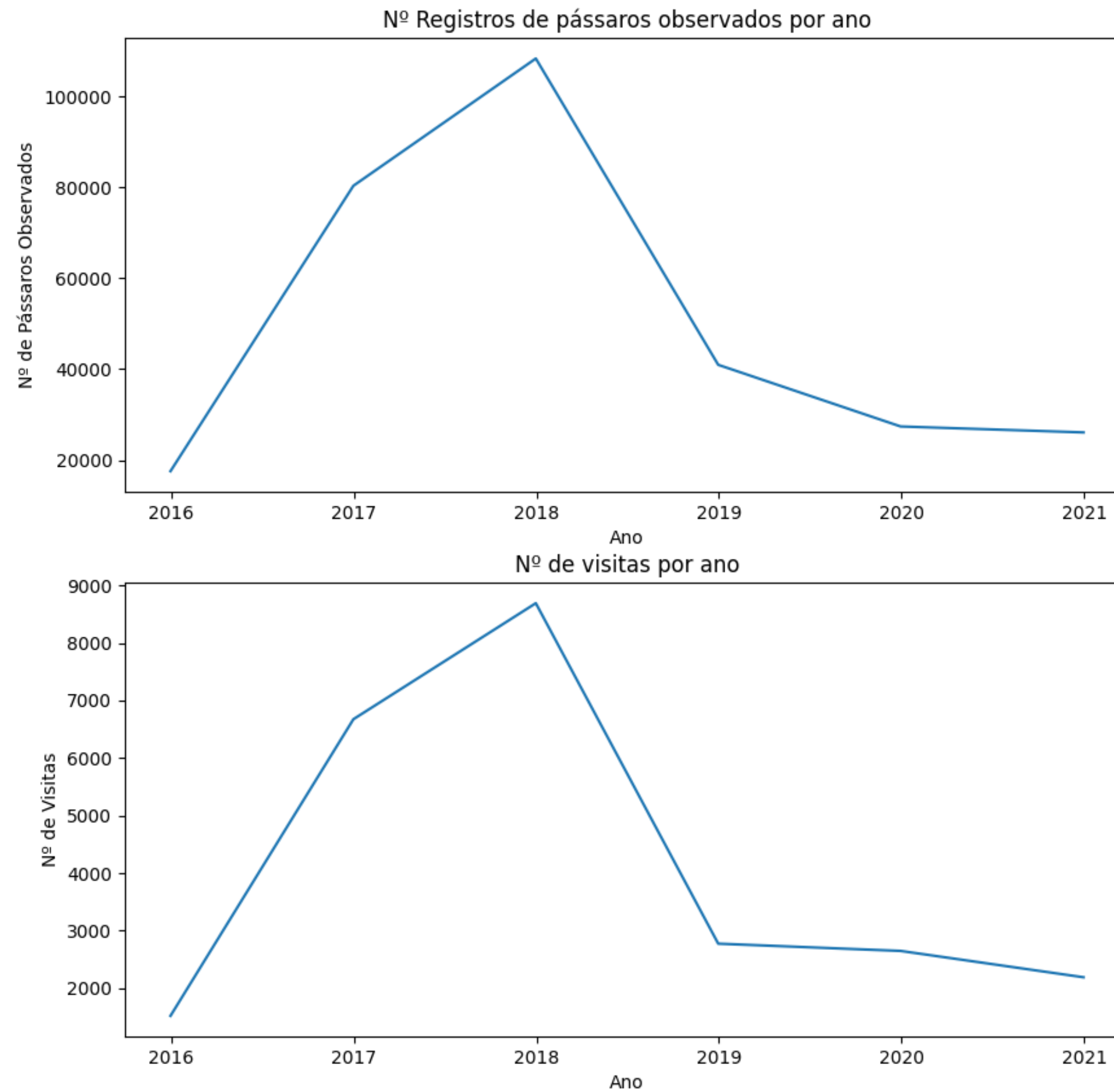
## Questionamentos

Como o tamanho do grupo afeta as observações?  
O número de espécies observados é alterado com o tempo?  
Como essa prática tem se comportado no período?

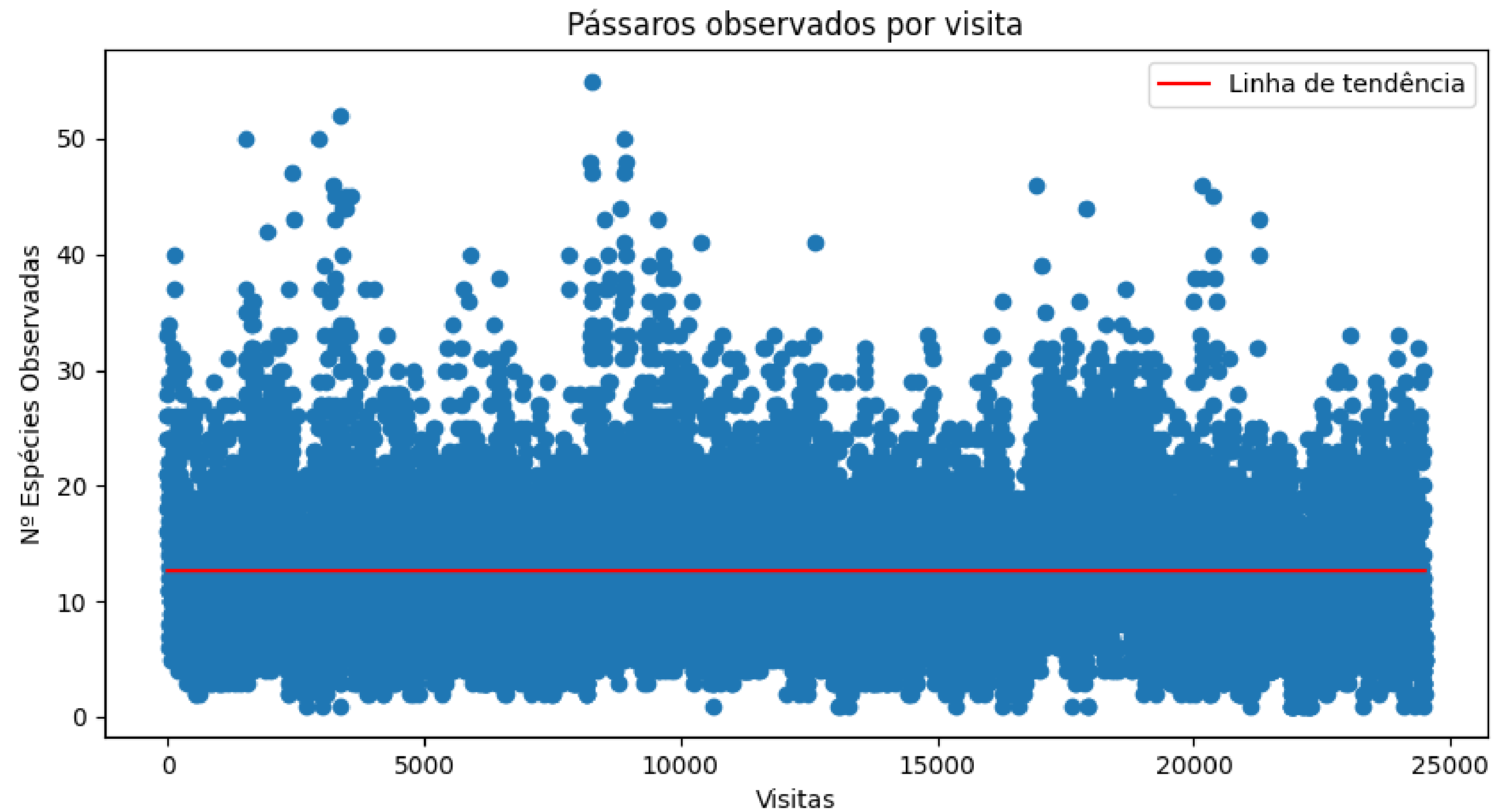
# Gráfico 01:



# Gráfico 02:



# Gráfico 03:



# Código:

```
# ----- PLOTANDO OS GRÁFICOS -----  
# Gráfico 01:  
plt.figure(figsize=(10,5))  
plt.title("Nº de Espécies Avistadas por Nº de observadores")  
z = np.polyfit(df_observacoes["Nº Observadores"], df_observacoes["Nº Espécies"], 1)  
p = np.poly1d(z)  
plt.plot(df_observacoes["Nº Observadores"], p(df_observacoes["Nº Observadores"]), "r", label='Linha de tendência')  
plt.scatter(df_observacoes["Nº Observadores"], df_observacoes["Nº Espécies"])  
plt.xlabel("Nº de Observadores")  
plt.ylabel("Nº de Espécies")  
plt.legend()  
plt.show()  
  
#Gráfico 02:  
plt.figure(figsize=(10,10))  
plt.subplot(2,1,1)  
plt.title("Nº Registros de pássaros observados por ano")  
plt.plot(df_passaros_ano.index, df_passaros_ano["Nº pássaros observados"])  
plt.xlabel("Ano")  
plt.ylabel("Nº de Pássaros Observados")  
  
plt.subplot(2, 1, 2)  
plt.title("Nº de visitas por ano")  
plt.plot(df_visitas_ano.index , df_visitas_ano["Nº Visitas"])  
plt.xlabel("Ano")
```



# 6. Insights:

## Observações relevantes:

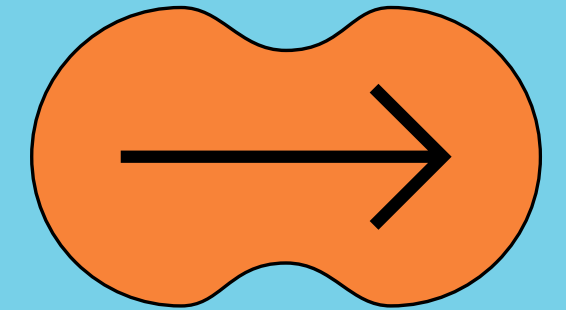
Houve uma queda abrupta no número de visitas para observação de pássaros em 2019, que tornou-se tendência.

Não há redução ou aumento do número de espécies observados por visita durante os cinco anos de dados coletados

Possivelmente há uma regra limitando a quantidade de visitantes por grupo, o que pode ser percebido por terem poucos registros de grupos maiores do que 8 pessoas

## Sugestão de ação:

Para maximizar a quantidade de pássaros avistados por grupo, prefira grupos de 2-5 integrantes



Alguma Dúvida?

