**EXPLORATORY DATA ANALYSIS OF FOREST FIRE IN BRAZIL**

****

* **Forest fires are a serious environmental threat that damage ecosystems, affect biodiversity, and contribute to climate change. Brazil, home to the Amazon rainforest — often called the "lungs of the Earth" — has witnessed a significant number of wildfires over the past few decades.**
* **This project aims to perform an Exploratory Data Analysis (EDA) of forest fire incidents in Brazil between 1998 and 2017.**
* **Using Python, Pandas, and visualization libraries like Matplotlib and Seaborn, this project uncovers seasonal trends, high-risk regions, and temporal patterns in the fire data to support environmental research and preventive planning**

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

import pandas as pd

# Read the CSV file from /content/ directory

data = pd.read\_csv('/content/amazon.csv', encoding='ISO-8859-1')

# Display first 5 rows of the dataset

print(data.head())

|  |
| --- |
| year state month number date |
| 0 1998 Acre Janeiro 0.0 1998-01-01 |
| 1 1999 Acre Janeiro 0.0 1999-01-01 |
| 2 2000 Acre Janeiro 0.0 2000-01-01 |
| 3 2001 Acre Janeiro 0.0 2001-01-01 |
| 4 2002 Acre Janeiro 0.0 2002-01-01 |

data.dtypes

print(data.dtypes)  # This will show the data types of each column.

year int64

state object

month object

number float64

date object

dtype: object

data['date']=pd.to\_datetime(data['date'])

print(data.dtypes)

year int64

state object

month object

number float64

date object

dtype: object

data.head()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| year | state | month | number | date |
| 0 | 1998 | Acre | Janeiro | 0.0 | 1998-01-01 |
| 1 | 1999 | Acre | Janeiro | 0.0 | 1999-01-01 |
| 2 | 2000 | Acre | Janeiro | 0.0 | 2000-01-01 |
| 3 | 2001 | Acre | Janeiro | 0.0 | 2001-01-01 |
| 4 | 2002 | Acre | Janeiro | 0.0 | 2002-01-01 |

data.shape

(6454, 5)

print("Number of rows",data.shape[0])

print("Number of coloumns",data.shape[1])

Number of rows 6454

Number of coloumns 5

**getting total about number of rows ,total number of coloumns ,Datatypes of each coloumn and Memory Requirement**

data.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 6454 entries, 0 to 6453

Data columns (total 5 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 year 6454 non-null int64

1 state 6454 non-null object

2 month 6454 non-null object

3 number 6454 non-null float64

4 date 6454 non-null object

dtypes: float64(1), int64(1), object(3)

memory usage: 252.2+ KB

**Check for duplicate data**

dup\_data=data.duplicated().any()

print(dup\_data)

True

data=data.drop\_duplicates()

data.shape

(6422, 5)

data.isnull().sum()

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | **0** | | --- | --- | | **year** | 0 | | **state** | 0 | | **month** | 0 | | **number** | 0 | | **date** | 0 |   dtype: int64 |

**get overall about the dataframe**

data.describe()

|  |  |
| --- | --- |
| year | number |
| count | 6422.000000 | 6422.000000 |
| mean | 2007.490969 | 108.815178 |
| std | 5.731806 | 191.142482 |
| min | 1998.000000 | 0.000000 |
| 25% | 2003.000000 | 3.000000 |
| 50% | 2007.000000 | 24.497000 |
| 75% | 2012.000000 | 114.000000 |
| max | 2017.000000 | 998.000000 |

data['date'] = pd.to\_datetime(data['date'])

data['date\_numeric'] = data['date'].astype('int64')  # Convert to Unix timestamp

# Use describe without errors

print(data.describe(include='all'))

|  |
| --- |
| year state month number \ |
| count 6422.000000 6422 6422 6422.000000 |
| unique NaN 23 12 NaN |
| top NaN Rio Setembro NaN |
| freq NaN 697 540 NaN |
| mean 2007.490969 NaN NaN 108.815178 |
| min 1998.000000 NaN NaN 0.000000 |
|  |
| 25% 2003.000000 NaN NaN 3.000000 |
| 50% 2007.000000 NaN NaN 24.497000 |
| 75% 2012.000000 NaN NaN 114.000000 |
| max 2017.000000 NaN NaN 998.000000 |
| std 5.731806 NaN NaN 191.142482 |
|  |

date date\_numeric

|  |
| --- |
| count 6422 6.422000e+03 |
| unique NaN NaN |
| top NaN NaN |
| freq NaN NaN |
| mean 2007-06-29 10:46:40.622859008 1.183114e+18 |
| min 1998-01-01 00:00:00 8.836128e+17 |
| 25% 2003-01-01 00:00:00 1.041379e+18 |
| 50% 2007-01-01 00:00:00 1.167610e+18 |
| 75% 2012-01-01 00:00:00 1.325376e+18 |
| max 2017-01-01 00:00:00 1.483229e+18 |
| std NaN 1.808828e+17 |

**Total number of fire registered ?**

data.shape

[ ]

data.shape

(6422, 6)

data.columns

Index(['year', 'state', 'month', 'number', 'date', 'date\_numeric'], dtype='object')

**in which the maximum number of forest fires were reported ?**

data.columns

Index(['year', 'state', 'month', 'number', 'date', 'date\_numeric'], dtype='object')

data1 = data.groupby('month')['number'].sum().reset\_index()

print(data1)

|  |
| --- |
| month number |
| 0 Abril 28184.770 |
| 1 Agosto 88050.435 |
| 2 Dezembro 57535.480 |
| 3 Fevereiro 30839.050 |
| 4 Janeiro 47681.844 |
| 5 Julho 92319.113 |
| 6 Junho 55997.675 |
| 7 Maio 34725.363 |
| 8 Março 30709.405 |
| 9 Novembro 85508.054 |
| 10 Outubro 88681.579 |
| 11 Setembro 58578.305 |

print(data1.columns)

Index(['month\_new', 'number'], dtype='object')

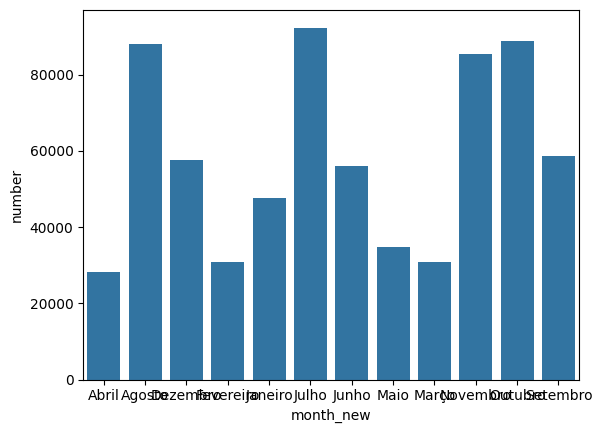
if 'month\_new' not in data.columns:

    data['month\_new'] = data['month']  # Assuming 'month' exists

data1 = data.groupby("month\_new")["number"].sum().reset\_index()

sns.barplot(x="month\_new", y="number", data=data1)

<Axes: xlabel='month\_new', ylabel='number'>

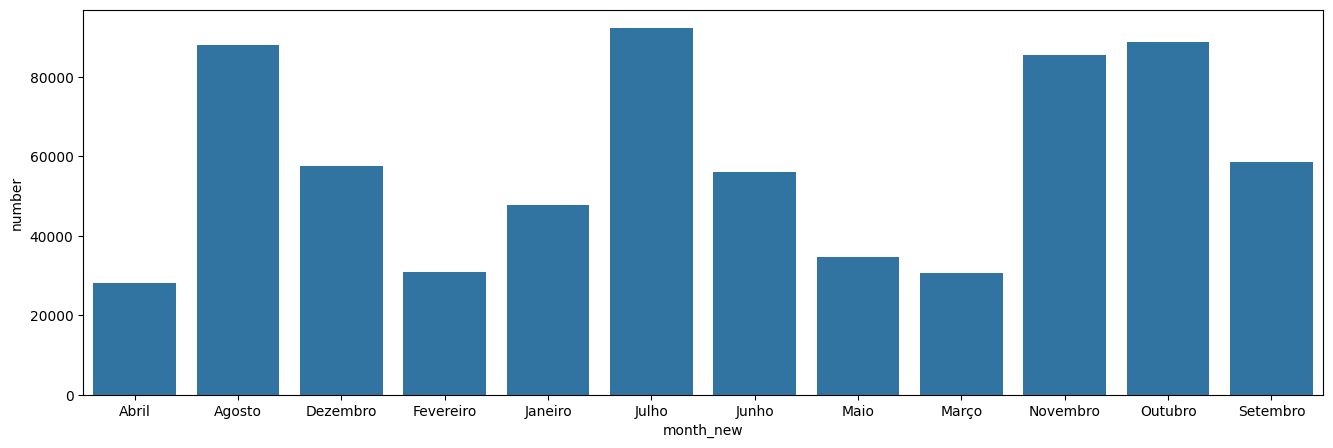


data['month\_new'] = data['month'].map({'Janeiro':'jan','Fevereiro':'Feb', 'Marco':'March', 'Abril':'April', 'Maio':'May', 'Junho':'June', 'Julho':'July','Agosto':'Aug','Septembro':'Sep','Outubro':'Oct','Novembro':'Nov','Dezembro':'Dec'})

plt.figure(figsize=(16,5))

sns.barplot(x="month\_new", y="number", data=data1)

<Axes: xlabel='month\_new', ylabel='number'>





**In which Yer Maximum Number Of forest Fires were Reported?**

data2 = data.groupby('year')['number'].sum().reset\_index()

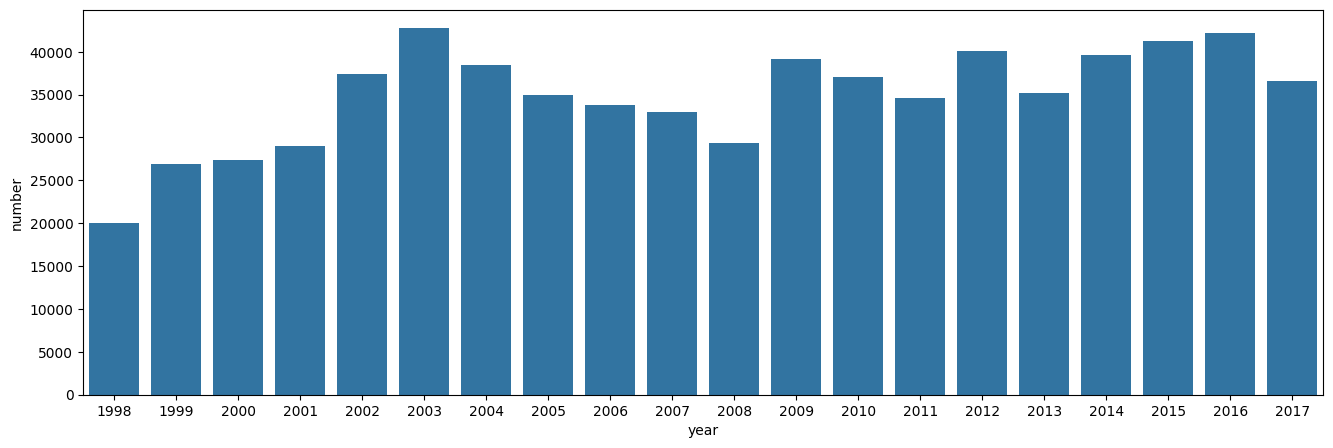
print(data1)

|  |
| --- |
|  |
| month\_new number |
| 0 Abril 28184.770 |
| 1 Agosto 88050.435 |
| 2 Dezembro 57535.480 |
| 3 Fevereiro 30839.050 |
| 4 Janeiro 47681.844 |
| 5 Julho 92319.113 |
| 6 Junho 55997.675 |
| 7 Maio 34725.363 |
| 8 Março 30709.405 |
| 9 Novembro 85508.054 |
| 10 Outubro 88681.579 |
| 11 Setembro 58578.305 |

plt.figure(figsize=(16,5))

sns.barplot(x="year",y="number",data=data2)

<Axes: xlabel='year', ylabel='number'>



**In which state the maximum number of fire were Reported?**

data.columns

Index(['year', 'state', 'month', 'number', 'date', 'date\_numeric','month\_new'],dtype='object')

data3=data.groupby('state')['number'].sum().reset\_index()data

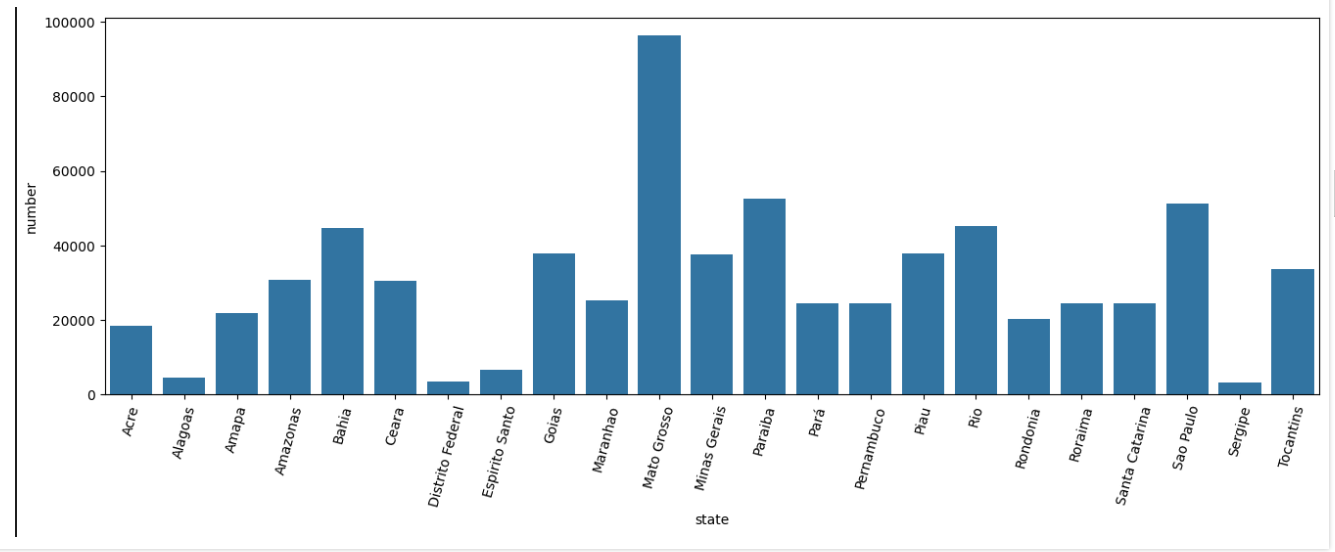
|  |  |
| --- | --- |
| state | number |
| 0 | Acre | 18464.030 |
| 1 | Alagoas | 4606.000 |
| 2 | Amapa | 21831.576 |
| 3 | Amazonas | 30650.129 |
| 4 | Bahia | 44746.226 |
| 5 | Ceara | 30428.063 |
| 6 | Distrito Federal | 3561.000 |
| 7 | Espirito Santo | 6546.000 |
| 8 | Goias | 37695.520 |
| 9 | Maranhao | 25129.131 |
| 10 | Mato Grosso | 96246.028 |
| 11 | Minas Gerais | 37475.258 |
| 12 | Paraiba | 52426.918 |
| 13 | Pará | 24512.144 |
| 14 | Pernambuco | 24498.000 |
| 15 | Piau | 37803.747 |
| 16 | Rio | 45094.865 |
| 17 | Rondonia | 20285.429 |
| 18 | Roraima | 24385.074 |
| 19 | Santa Catarina | 24359.852 |
| 20 | Sao Paulo | 51121.198 |
| 21 | Sergipe | 3237.000 |
| 22 | Tocantins | 33707.885 |

plt.figure(figsize=(16,5))

sns.barplot(x="state",y="number",data=data3)

plt.xticks(rotation=75)

plt.show()



**Find the total number of Fires were Reported in Amazons**

data.columns

Index(['year', 'state', 'month', 'number', 'date', 'date\_numeric', 'month\_new'],dtype='object')

data[data['state']=='Amazonas']['number'].sum()

np.float64(30650.129)

**Display the fires were Reported in Amazonas**

data.columns

Index(['year', 'state', 'month', 'number', 'date', 'date\_numeric', 'month\_new'] ,dtype='object')

data4=data[data['state']=='Amazonas']

data5=data4.groupby('year')['number'].sum().reset\_index()

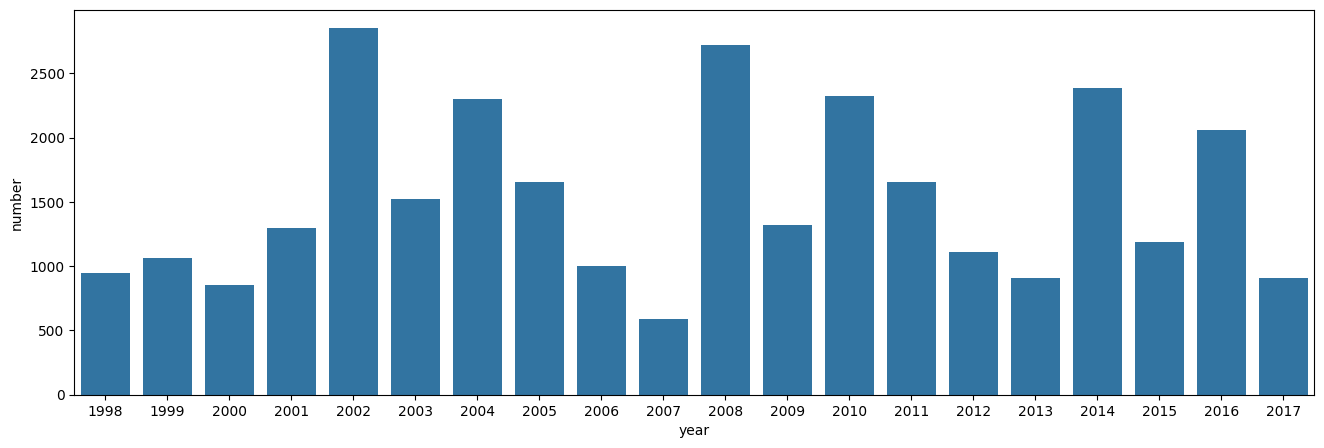
data5

|  |  |
| --- | --- |
| year | number |
| 0 | 1998 | 946.000 |
| 1 | 1999 | 1061.000 |
| 2 | 2000 | 853.000 |
| 3 | 2001 | 1297.000 |
| 4 | 2002 | 2852.000 |
| 5 | 2003 | 1524.268 |
| 6 | 2004 | 2298.207 |
| 7 | 2005 | 1657.128 |
| 8 | 2006 | 997.640 |
| 9 | 2007 | 589.601 |
| 10 | 2008 | 2717.000 |
| 11 | 2009 | 1320.601 |
| 12 | 2010 | 2324.508 |
| 13 | 2011 | 1652.538 |
| 14 | 2012 | 1110.641 |
| 15 | 2013 | 905.217 |
| 16 | 2014 | 2385.909 |
| 17 | 2015 | 1189.994 |
| 18 | 2016 | 2060.972 |
| 19 | 2017 | 906.905 |

plt.figure(figsize=(16,5))

sns.barplot(x="year",y="number",data=data5)

<Axes: xlabel='year', ylabel='number'>



**Display the fires were Reported in Amazonas (day wise)**

data.columns

Index(['year', 'state', 'month', 'number', 'date', 'date\_numeric','month\_new'] dtype='object')

data6=data[data['state']=='Amazonas']

**Find the total number of Fires were Reported in 2015 And Visulize data Baesd on Each Month**

data.columns

Index(['year', 'state', 'month', 'number', 'date', 'date\_numeric','month\_new'] dtype='object')

fire=data[data['year']==2015].groupby('month\_new')['number'].sum().reset\_index()

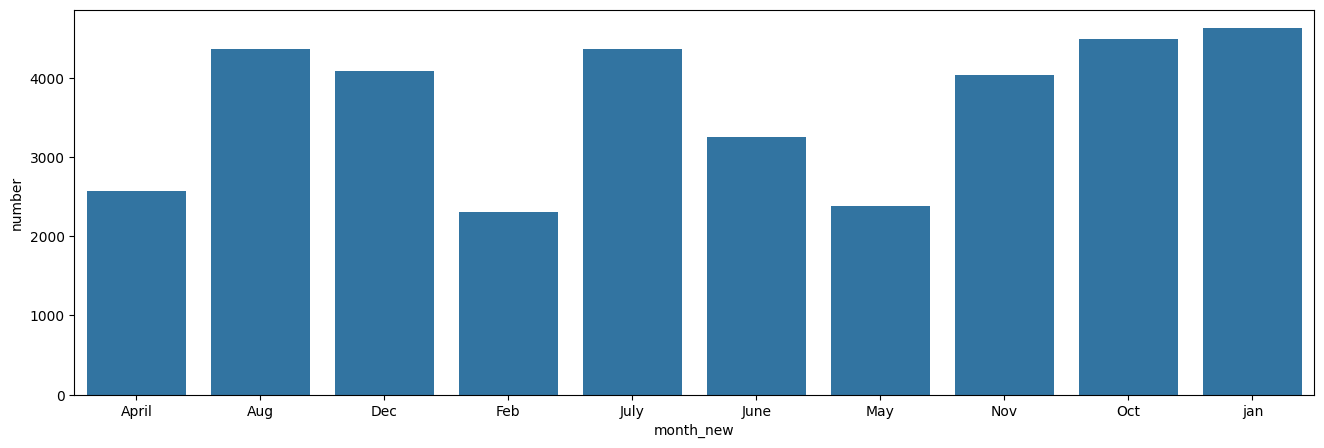
fire

|  |  |
| --- | --- |
| month\_new | number |
| 0 | April | 2573.000 |
| 1 | Aug | 4363.125 |
| 2 | Dec | 4088.522 |
| 3 | Feb | 2309.000 |
| 4 | July | 4364.392 |
| 5 | June | 3260.552 |
| 6 | May | 2384.000 |
| 7 | Nov | 4034.518 |
| 8 | Oct | 4499.525 |
| 9 | jan | 4635.000 |

plt.figure(figsize=(16,5))

sns.barplot(x="month\_new",y="number",data=fire)

<Axes: xlabel='month\_new', ylabel='number'>



**Find the average number of fires were reported from Highest to lowest (state-wise)**

data.columns

Index(['year', 'state', 'month', 'number', 'date', 'date\_numeric',

'month\_new'],

dtype='object')

data8=data.groupby('state')['number'].mean().sort\_values(ascending=False).reset\_index()

data8

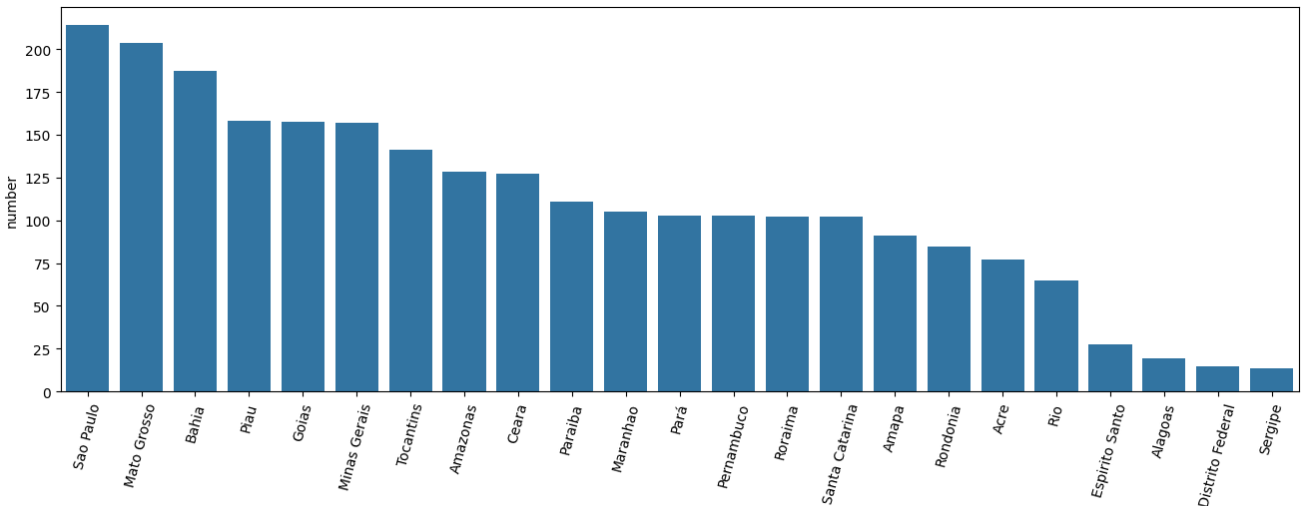
|  |  |
| --- | --- |
| state | number |
| 0 | Sao Paulo | 213.896226 |
| 1 | Mato Grosso | 203.479975 |
| 2 | Bahia | 187.222703 |
| 3 | Piau | 158.174674 |
| 4 | Goias | 157.721841 |
| 5 | Minas Gerais | 156.800243 |
| 6 | Tocantins | 141.037176 |
| 7 | Amazonas | 128.243218 |
| 8 | Ceara | 127.314071 |
| 9 | Paraiba | 111.073979 |
| 10 | Maranhao | 105.142808 |
| 11 | Pará | 102.561272 |
| 12 | Pernambuco | 102.502092 |
| 13 | Roraima | 102.029598 |
| 14 | Santa Catarina | 101.924067 |
| 15 | Amapa | 91.345506 |
| 16 | Rondonia | 84.876272 |
| 17 | Acre | 77.255356 |
| 18 | Rio | 64.698515 |
| 19 | Espirito Santo | 27.389121 |
| 20 | Alagoas | 19.271967 |
| 21 | Distrito Federal | 14.899582 |
| 22 | Sergipe | 13.543933 |

plt.figure(figsize=(16,5))

sns.barplot(x="state",y="number",data=data8)

plt.xticks(rotation=75)

plt.show()



data[data['month\_new']=="Dec"]['state'].unique()

array(['Acre', 'Alagoas', 'Amapa', 'Amazonas', 'Bahia', 'Ceara',

'Distrito Federal', 'Espirito Santo', 'Goias', 'Maranhao',

'Mato Grosso', 'Minas Gerais', 'Pará', 'Paraiba', 'Pernambuco',

'Piau', 'Rio', 'Rondonia', 'Roraima', 'Santa Catarina',

'Sao Paulo', 'Sergipe', 'Tocantins'], dtype=object)

✅ **Q1: How is this project helpful in real-world life/environment?**

**✔️ This project helps environmental researchers, forest authorities, and policymakers understand when and where forest fires are most likely to occur. By identifying high-risk months and regions (e.g., Mato Grosso in July), preventive actions like resource allocation, early warnings, and public awareness campaigns can be planned more effectively. It supports real-world climate action, disaster management, and sustainable forestry efforts**.

✅ **Q2: How is this project helpful to all of us?**

**✔️ Forest fires have a direct impact on biodiversity, climate change, and human health. By analyzing forest fire trends over time, this project increases awareness of environmental risks and encourages data-driven decisions to protect our forests. It highlights the importance of using data science for social good, enabling citizens, researchers, and governments to collaborate for a safer and greener future**

****