



# Incêndios Florestais no Brasil

CONSCIENTIZAÇÃO SOBRE OS IMPACTOS AMBIENTAIS DAS QUEIMADAS

**Grupo:**

Samuel

Victor César Sampaio Ferreira da Silva

Gilmar

Breno

Jonathan

**Disciplina:** Tópicos de big data em Python / Estácio

**Professor:** Davi barros

# Introdução

- ▶ As **queimadas no Brasil** representam um dos maiores desafios ambientais do país, com consequências devastadoras para a biodiversidade, o clima e a saúde humana. Historicamente associadas à abertura de novas áreas para a agropecuária, ao manejo inadequado do solo e a fatores climáticos como secas prolongadas, as queimadas se intensificaram nos últimos anos, especialmente em biomas como a Amazônia, o Cerrado e o Pantanal.
- ▶ As florestas são pilares fundamentais para a manutenção do equilíbrio ambiental global. Elas atuam como verdadeiros "pulmões do mundo", absorvendo dióxido de carbono e liberando oxigênio, um processo vital para a regulação climática e a qualidade do ar que respiramos. Além disso, desempenham um papel crucial na regulação do ciclo hidrológico, influenciando regimes de chuva e prevenindo secas e inundações.

O fogo que destrói a floresta também ameaça nosso futuro.

# Objetivo

- ▶ Nosso projeto busca **conscientizar** a população sobre os graves impactos ambientais das queimadas. Para isso, vamos além de apenas informar; queremos utilizar dados reais para demonstrar a verdadeira gravidade do problema, tornando os efeitos visíveis e inegáveis.
- ▶ Além disso, nosso objetivo é incentivar atitudes sustentáveis e apoiar políticas de preservação que promovam um futuro mais seguro e equilibrado para todos. Queremos transformar a preocupação em ação, mostrando que cada um pode fazer a diferença na proteção de nossas florestas.

# Metodologia

- ▶ A coleta dos dados foram feitas no site do INPE ( Instituto Nacional de Pesquisas Espaciais), ano 2021 a 2024.
- ▶ Usamos Python para limpar e tratar os dados usando a biblioteca **Pandas**
- ▶ Usamos análise estatísticas e identificações de padrões
- ▶ Utilizamos Matplotlib e Seaborn para geração de gráficos e visualizações dos dados
- ▶ Ferramentas:
  - Linguagem: Python
  - Bibliotecas: Pandas, Matplotlib, Seaborn
- ▶ Fonte de dados: <https://terrabrasilis.dpi.inpe.br/queimadas/portal/dados-abertos/#da-rf>
- ▶ [https://dataserver-coids.inpe.br/queimadas/queimadas/focos/csv/anual/Brasil\\_todos\\_sats/](https://dataserver-coids.inpe.br/queimadas/queimadas/focos/csv/anual/Brasil_todos_sats/)
- ▶ <https://terrabrasilis.dpi.inpe.br/app/dashboard/fires/biomes/aggregated/>
- ▶ <https://www.gov.br/cemaden/pt-br/assuntos/monitoramento/monitoramento-de-seca-para-o-brasil>
- ▶ <https://www.kaggle.com/datasets/gnomows/dados-metereologicos-2018-2024-inmet/data>
- ▶ <https://bdmep.inmet.gov.br/>
- ▶ Nosso foco foi identificar as regiões e períodos mais críticos, buscando relacionar dados com os efeitos ambientais das queimadas

# Dados utilizados no projeto

- ▶ **Período Analisado:** de 2021 a 2024
- ▶ Fonte de dados: Banco de dados do INPE ( Instituto Nacional de Pesquisas Espaciais ).
- ▶ Informações coletadas:
  - ❑ Quantidade de focos de queimadas por estado e por mês
  - ❑ Distribuição ao longo dos anos
  - ❑ Regiões mais afetadas
  - ❑ Picos de ocorrência por período
- ▶ Os dados são públicos e atualizados constantemente
- ▶ A base de dados nos permitiu identificar padrões e tendências importantes para nossa análise.

```

anos = [2021, 2022, 2023, 2024]

# Dicionário: {estado: {ano: ocorrências}}
dados = {}

for ano in anos:
    try:
        df = pd.read_csv(f'data/focos_br_todos-sats_{ano}.csv', usecols=['estado'])
        contagem = df['estado'].value_counts()
        for estado, qtd in contagem.items():
            if estado not in dados:
                dados[estado] = {}
            dados[estado][ano] = qtd
    except Exception as e:
        print(f"Erro no ano {ano}: {e}")

# Transforma o dicionário em DataFrame
df_estados = pd.DataFrame(dados).fillna(0).astype(int)
df_estados = df_estados.T # Estados nas linhas, anos nas colunas

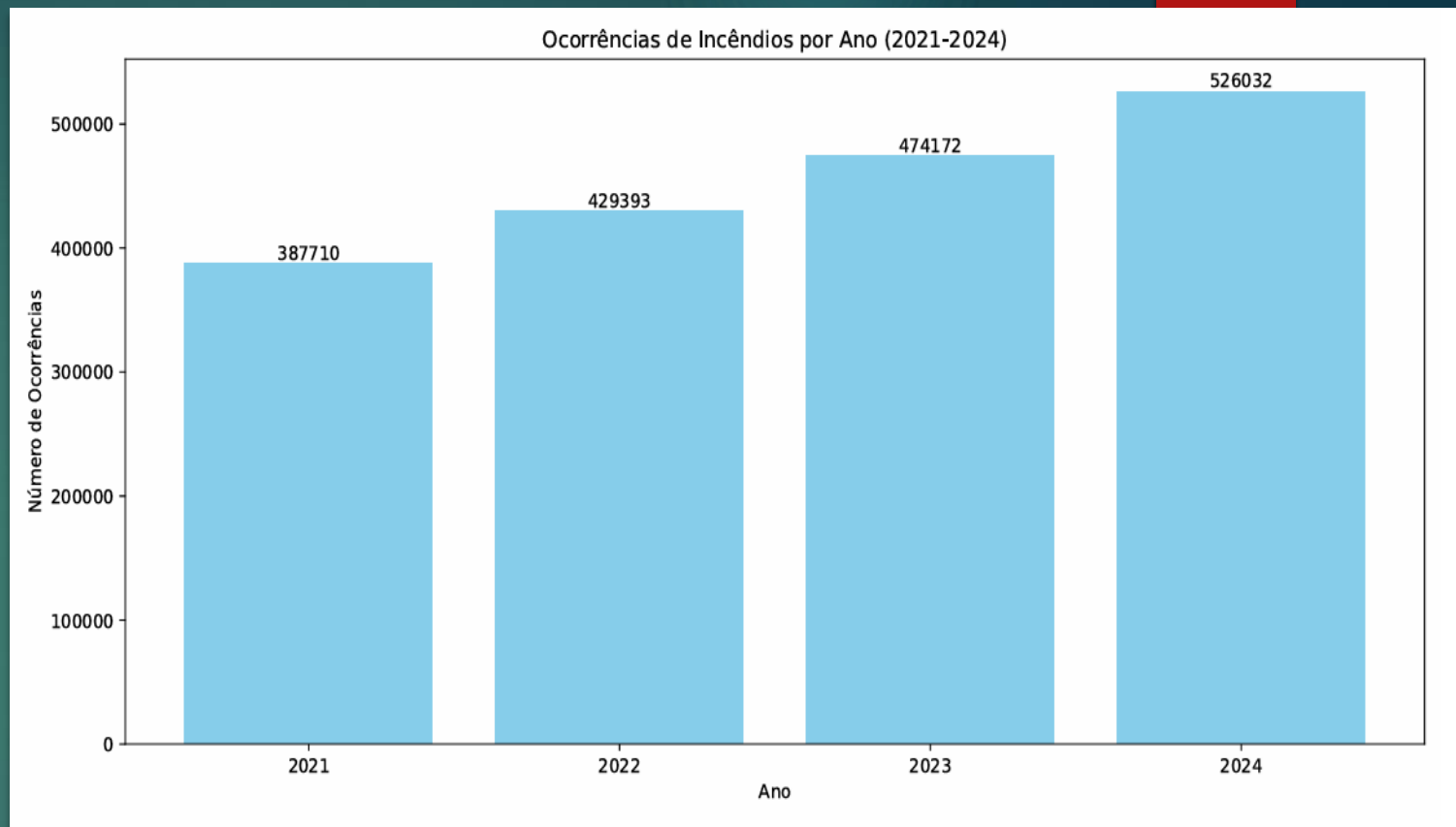
# Soma total de ocorrências por estado para selecionar os top 10
top_10_estados = df_estados.sum(axis=1).sort_values(ascending=False).head(10).index
df_top10 = df_estados.loc[top_10_estados]

# Transpõe para ter os anos no eixo X e os estados como colunas
df_plot = df_top10.T

# Plotando o gráfico
plt.figure(figsize=(12, 7))
for estado in df_plot.columns:
    plt.plot(df_plot.index, df_plot[estado], marker='o', label=estado)

plt.title('Top 10 Estados com Mais Ocorrências de Incêndios por Ano')
plt.xlabel('Ano')
plt.ylabel('Número de Ocorrências')
plt.legend(title='Estado', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.grid(True)
plt.tight_layout()
plt.savefig('Top10_ocorrencias_por_estado_ano.pdf')
plt.close()

```



# Evolução das queimadas no Brasil



```

for ano in anos:
    try:
        df = pd.read_csv(f'data/focos_br_todos-sats_{ano}.csv', usecols=['estado'])
        contagem = df['estado'].value_counts()
        for estado, qtd in contagem.items():
            if estado not in dados:
                dados[estado] = {}
            dados[estado][ano] = qtd
    except Exception as e:
        print(f"Erro no ano {ano}: {e}")

# Transforma o dicionário em DataFrame
df_estados = pd.DataFrame(dados).fillna(0).astype(int)
df_estados = df_estados.T # Estados nas linhas, anos nas colunas

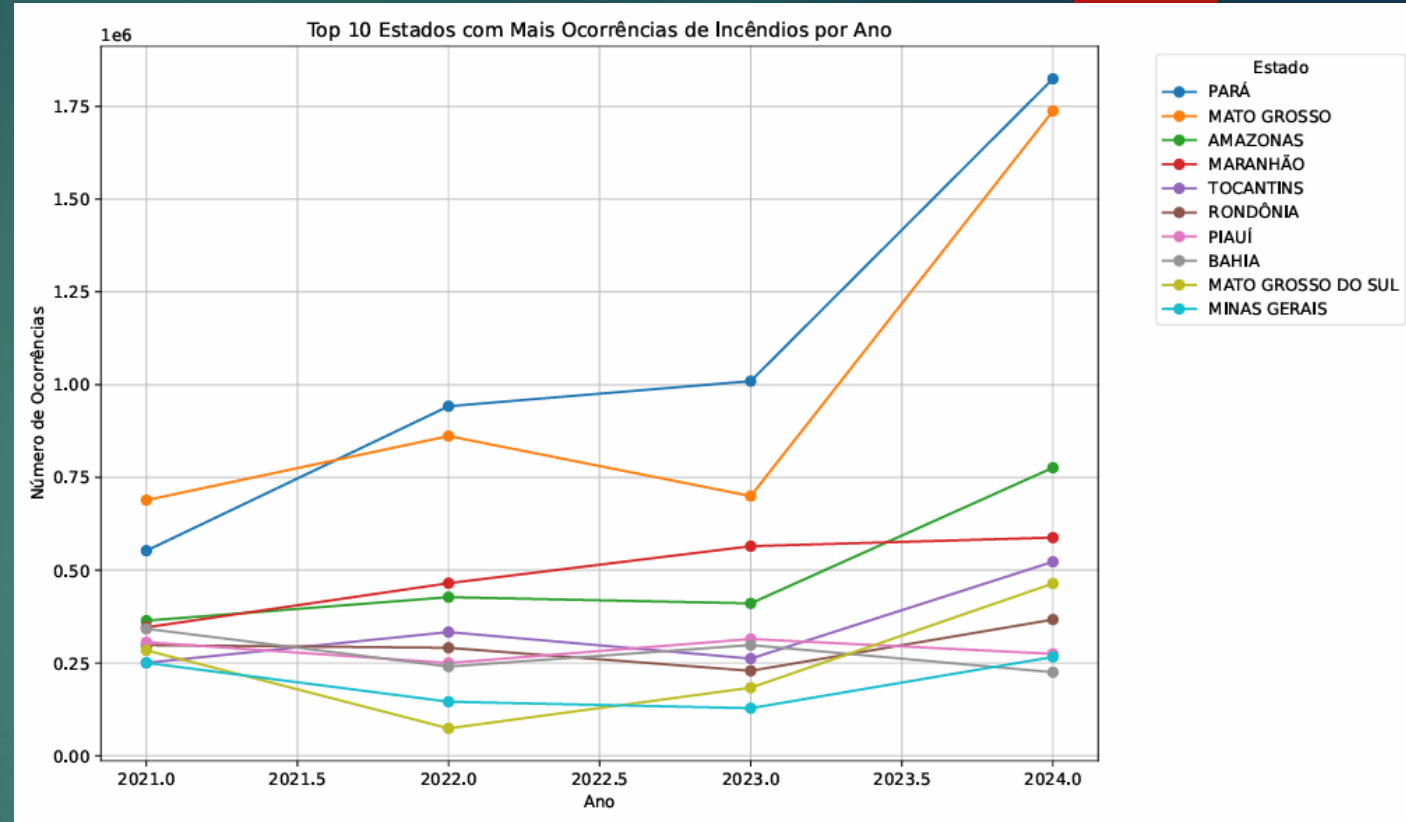
# Soma total de ocorrências por estado para selecionar os top 10
top_10_estados = df_estados.sum(axis=1).sort_values(ascending=False).head(10).index
df_top10 = df_estados.loc[top_10_estados]

# Transpõe para ter os anos no eixo X e os estados como colunas
df_plot = df_top10.T

# Plotando o gráfico
plt.figure(figsize=(12, 7))
for estado in df_plot.columns:
    plt.plot(df_plot.index, df_plot[estado], marker='o', label=estado)

plt.title('Top 10 Estados com Mais Ocorrências de Incêndios por Ano')
plt.xlabel('Ano')
plt.ylabel('Número de Ocorrências')
plt.legend(title='Estado', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.grid(True)
plt.tight_layout()
plt.savefig('Top10_ocorrencias_por_estado_ano.pdf')
plt.close()

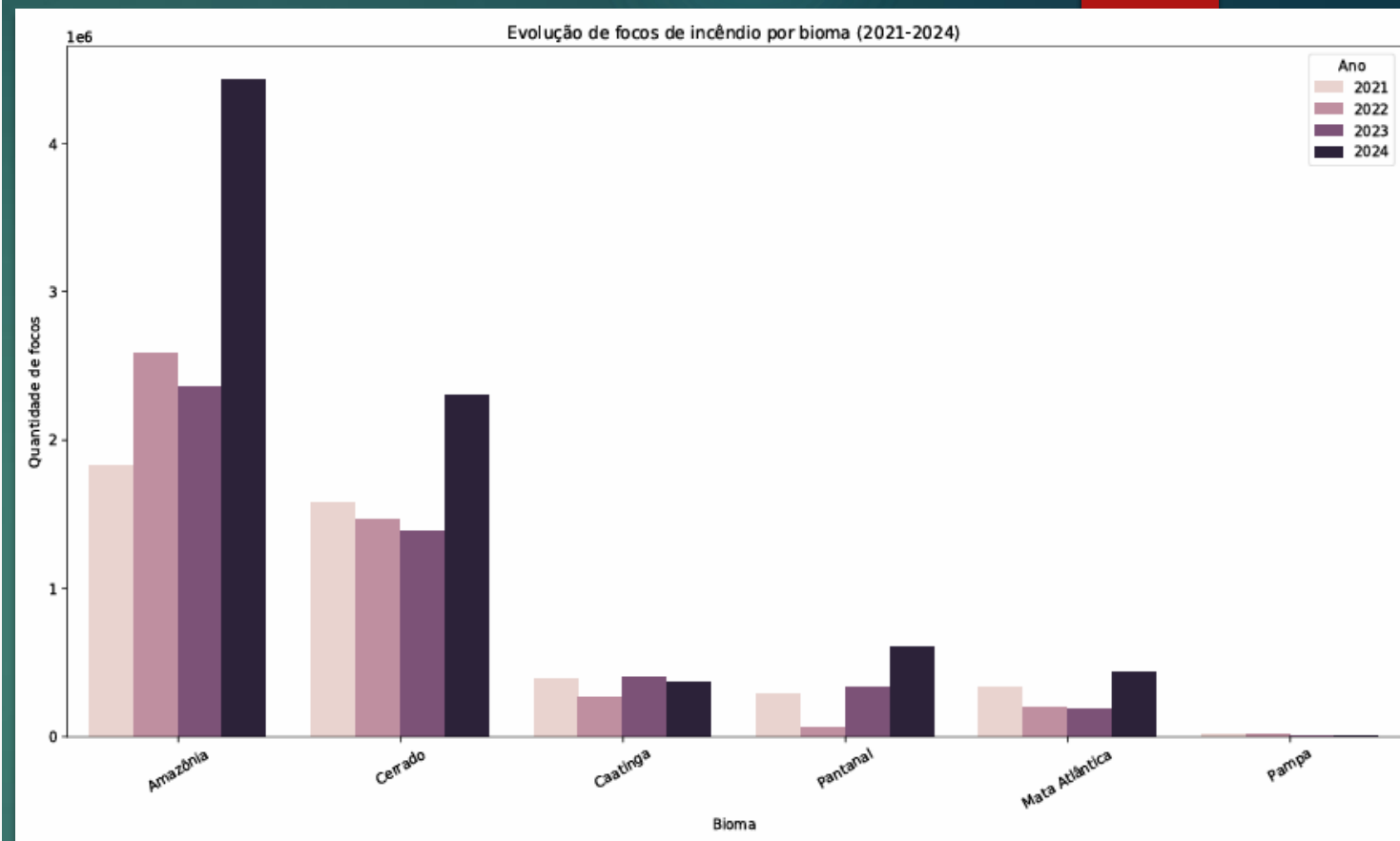
```



# Evolução das queimadas no Brasil por estado

and Inserir célula de código abaixo (Ctrl+M B)

```
def analise_de_queimadas_por_bioma(anos):  
    # Lista para armazenar DataFrames de cada ano  
    lista_dfs = []  
  
    for ano in anos:  
        try:  
            df = pd.read_csv(f'data/focos_br_todos-sats_{ano}.csv', usecols=['bioma'])  
            df = df.dropna(subset=['bioma']) # remove valores nulos  
            df['ano'] = ano # adiciona coluna ano  
            lista_dfs.append(df)  
        except Exception as e:  
            print(f"Erro ao ler {ano}: {e}")  
  
    if not lista_dfs:  
        print("Nenhum dado carregado. Verifique os arquivos CSV.")  
        return  
  
    # Concatenar todos os DataFrames  
    df_completo = pd.concat(lista_dfs, ignore_index=True)  
  
    # Ordenar biomas pela contagem total  
    ordem_biomas = df_completo['bioma'].value_counts().index  
  
    # Gráfico com colunas separadas por ano  
    plt.figure(figsize=(14, 8))  
    sns.countplot(data=df_completo, x='bioma', hue='ano', order=ordem_biomas)  
    plt.title('Evolução de focos de incêndio por bioma (2021-2024)')  
    plt.ylabel('Quantidade de focos')  
    plt.xlabel('Bioma')  
    plt.xticks(rotation=30)  
    plt.legend(title='Ano')  
    plt.tight_layout()  
    plt.savefig('evolucao_biomas_anos_2021_-_2024.pdf', format='pdf')  
    plt.close()  
  
# Executar a função  
analise_de_queimadas_por_bioma(anos)
```



# Evolução das queimadas no Brasil por bioma



```

desmatamento = df[df['class'].str.contains('desmatamento', case=False)] # Filtra linhas cujo 'class' contém 'desmatamento'

# Agregações dos focos totais e em áreas desmatadas, por estado

# Total de focos por estado (independente da classe)
focos_total_estado = df.groupby('uf_sigla')['focos'].sum().reset_index()
focos_total_estado = focos_total_estado.rename(columns={'focos': 'Total de Focos'})

# Total de focos apenas nas áreas de desmatamento
focos_desmatamento_estado = desmatamento.groupby('uf_sigla')['focos'].sum().reset_index()
focos_desmatamento_estado = focos_desmatamento_estado.rename(columns={'focos': 'Focos em Desmatamento'})

#juntar dois DataFrames para comparação lado a lado

comparativo = pd.merge(focos_total_estado, focos_desmatamento_estado, on='uf_sigla', how='left')
comparativo = comparativo.fillna(0) # Substitui possíveis NaNs por 0
comparativo = comparativo.sort_values('Total de Focos', ascending=False) # Ordena pelos maiores focos totais

# Prepara os dados em formato long (necessário para seaborn agrupar por 'Tipo')
comparativo_melt = comparativo.melt(
    id_vars='uf_sigla',
    value_vars=['Total de Focos', 'Focos em Desmatamento'],
    var_name='Tipo',
    value_name='Focos'
)

#Criação do gráfico de barras agrupadas

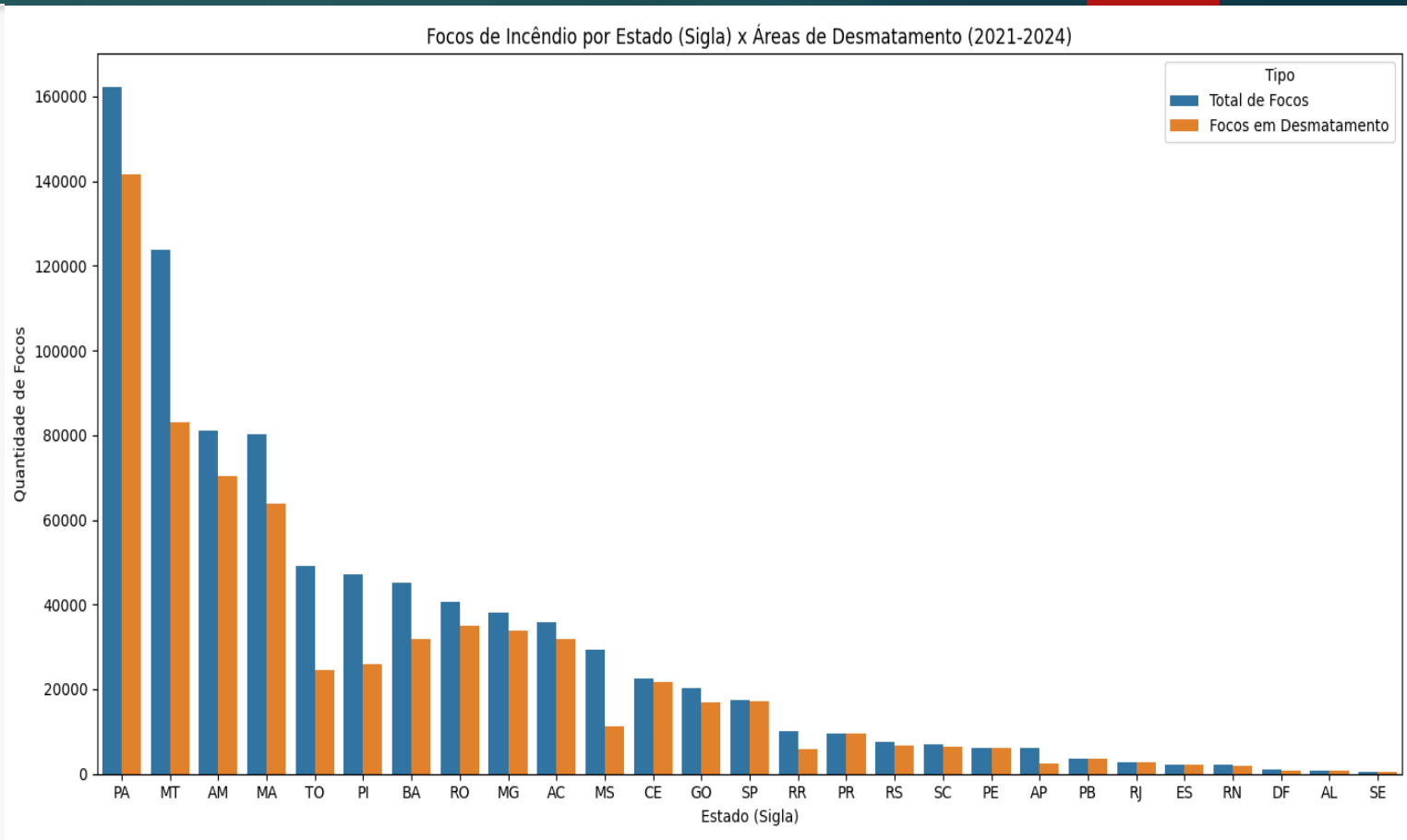
plt.figure(figsize=(14, 7))
sns.barplot(data=comparativo_melt, x='uf_sigla', y='Focos', hue='Tipo')

plt.title('Focos de Incêndio por Estado (Sigla) x Áreas de Desmatamento (2021-2024)')
plt.xlabel('Estado (Sigla)')
plt.ylabel('Quantidade de Focos')
plt.legend(title='Tipo')

plt.tight_layout()

plt.savefig('graficos/Focos de Incêndio por Estado (Sigla) x Áreas de Desmatamento.png')
plt.close()

```



# Focos de incêndios por estado x áreas desmatadas

```

# Padroniza os nomes das colunas: remove espaços, converte para minúsculas
df.columns = df.columns.str.strip().str.lower()

# Padroniza os nomes dos estados para letras maiúsculas
df['uf'] = df['uf'].str.upper()

# Converte os nomes dos estados para siglas, usando o dicionário acima
# Se já estiver em sigla, mantém como está
df['uf_sigla'] = df['uf'].map(estado_para_sigla).fillna(df['uf'])

# Converte a coluna 'focos' para número (caso haja valores inválidos, converte para NaN)
df['focos'] = pd.to_numeric(df['focos'], errors='coerce')

# Converte a coluna de data para o tipo datetime. Espera o formato 'YYYY/MM'
df['date'] = pd.to_datetime(df['date'], format='%Y/%m', errors='coerce')

# Filtra apenas os registros cuja classe indica desmatamento
desmatamento = df[df['class'].str.contains('desmatamento', case=False)]

# Agrupa os dados totais por estado (sigla), somando o número de focos
total_focos = df.groupby('uf_sigla')['focos'].sum().reset_index(name='Total de Focos')

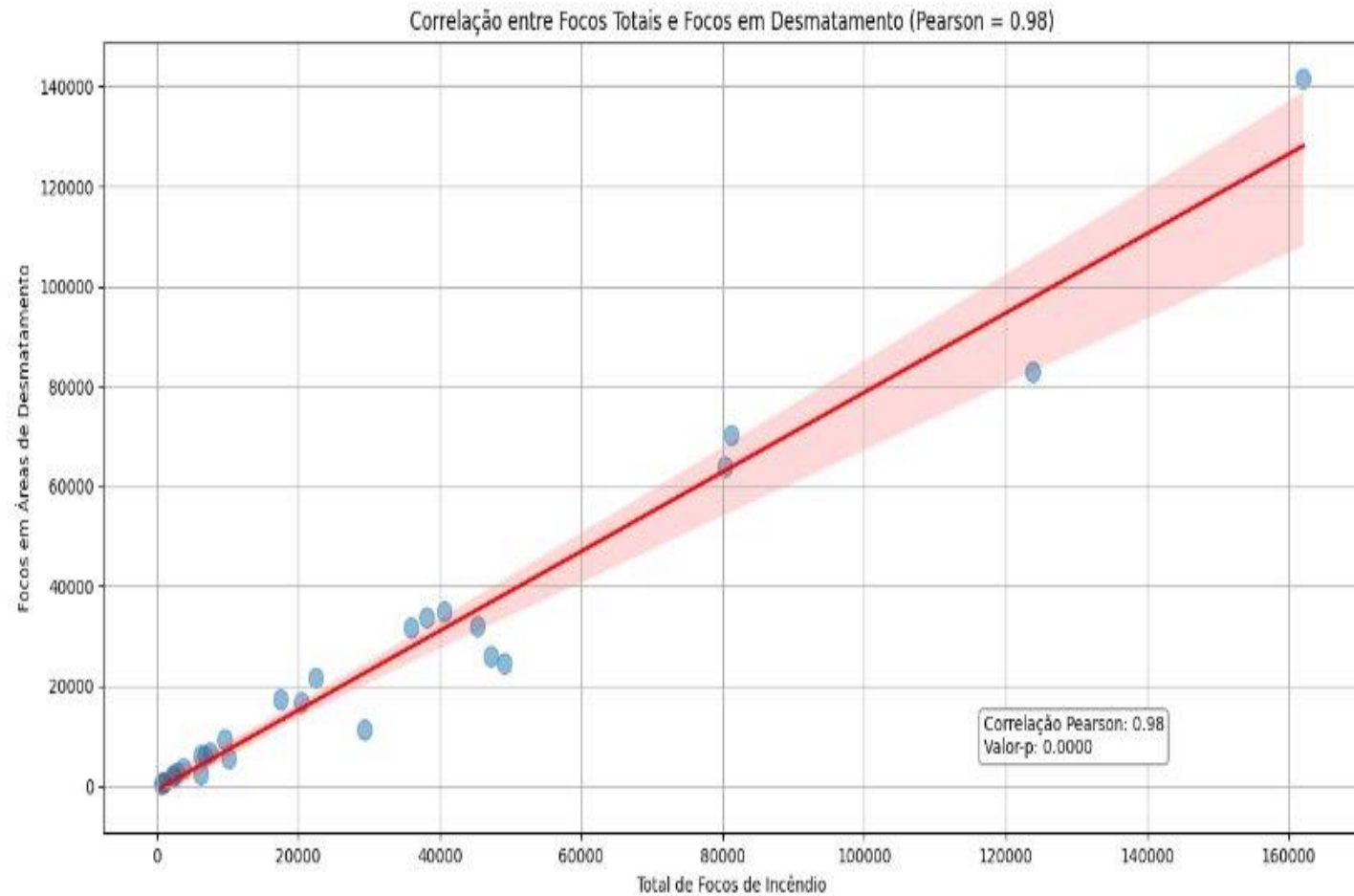
# Agrupa os focos relacionados a desmatamento por estado, somando também
focos_desmat = desmatamento.groupby('uf_sigla')['focos'].sum().reset_index(name='Focos em Desmatamento')

# Junta os dois DataFrames em um só, com base na sigla dos estados
# Usa left join para manter todos os estados que aparecem em total_focos
comparativo = pd.merge(total_focos, focos_desmat, on='uf_sigla', how='left').fillna(0)

# Calcula a correlação de Pearson entre focos totais e focos em desmatamento
correlacao, p_valor = pearsonr(comparativo['Total de Focos'], comparativo['Focos em Desmatamento'])

# Cria um gráfico de dispersão com linha de regressão entre os dois conjuntos de dados
plt.figure(figsize=(10, 6))
sns.regplot(
    data=comparativo,
    x='Total de Focos',
    y='Focos em Desmatamento',
    scatter_kws={'s': 80, 'alpha': 0.6}, # Personaliza os pontos do gráfico
    line_kws={'color': 'red'} # Cor da linha de regressão
)

```



# Focos de incêndios x desmatamentos

```

# Junta os dados de incêndios com os dados de clima, pela chave (estado, ano_mes)
df = pd.merge(incendios_agg, clima_agg, on=['estado', 'ano_mes'], how='inner')

# Junta o resultado anterior com os dados de desmatamento
df = pd.merge(df, desmatamento_pivot, on=['estado', 'ano_mes'], how='left')

# Remove registros com valores nulos após as junções
df = df.dropna()

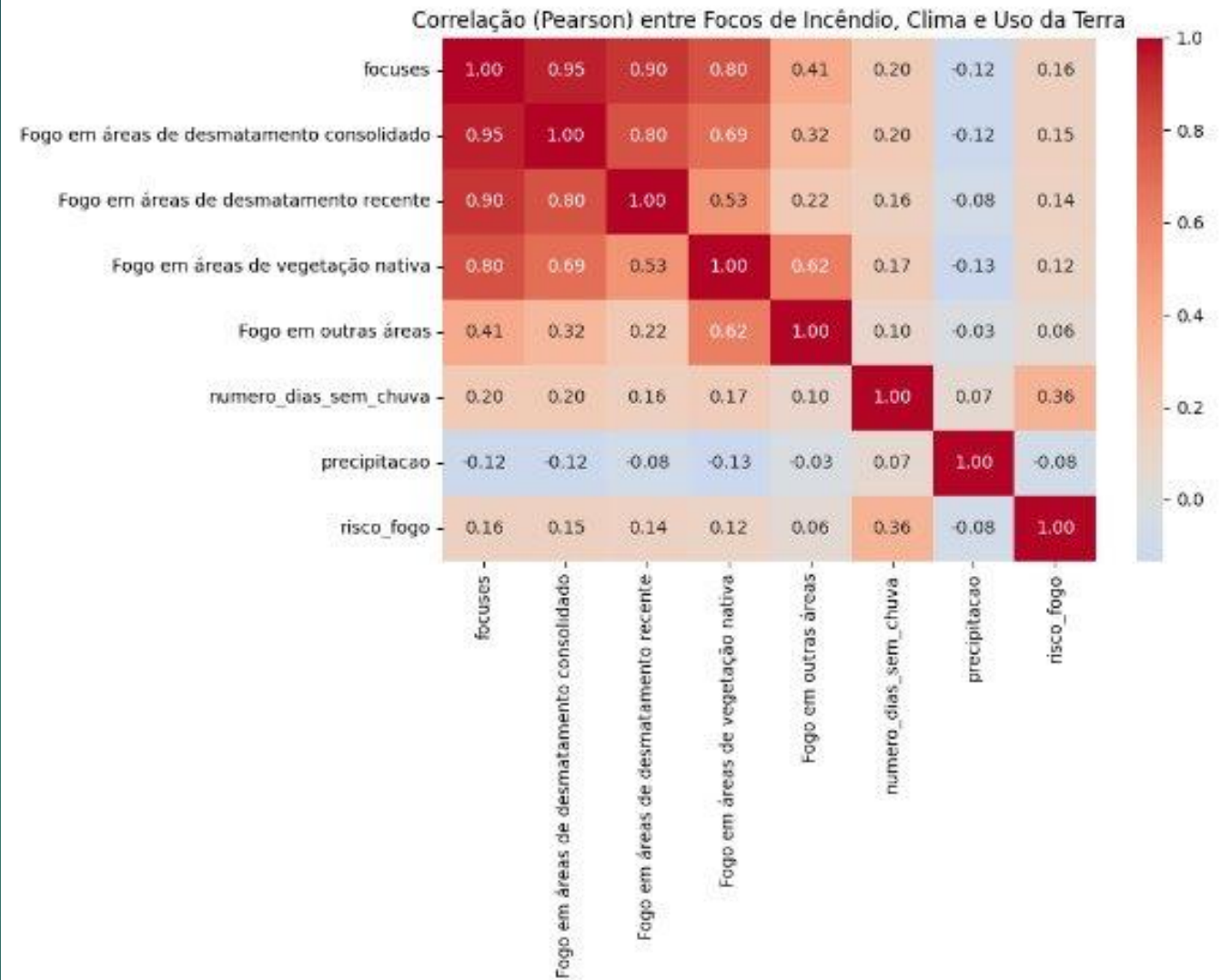
# Lista de variáveis que serão incluídas na análise de correlação
variaveis_relevantes = [
    'focos', # Total de focos de incêndio
    'Fogo em áreas de desmatamento consolidado',
    'Fogo em áreas de desmatamento recente',
    'Fogo em áreas de vegetação nativa',
    'Fogo em outras áreas',
    'numero_dias_sem_chuva',
    'precipitacao',
    'risco_fogo'
]

# Calcula a matriz de correlação de Pearson entre as variáveis selecionadas
df_corr_relevantes = df[variaveis_relevantes].corr(method='pearson')

# Criação do heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(
    df_corr_relevantes, annot=True, cmap='coolwarm', fmt=".2f", center=0
)
plt.title('Correlação (Pearson) entre Focos de Incêndio, Clima e Uso da Terra')
plt.tight_layout()

# Salva o gráfico como imagem
plt.savefig('graficos/Correlação entre Focos de Incêndio, Clima e Uso da Terra.png')
plt.close()

```



# Correlação entre focos x desmatamento x fatores climaticos

# Conclusão

- ▶ A análise dos dados entre 2021 e 2024 revelou que os focos de incêndio no Brasil continuam fortemente associados ao desmatamento, especialmente em biomas como a Amazônia e o Cerrado. Identificamos que os estados com maior número de focos coincidem com regiões críticas de desmatamento, e fatores climáticos como seca prolongada e aumento do risco de fogo também contribuem significativamente para a intensificação das queimadas.
- ▶ Nossos gráficos e análises estatísticas demonstraram correlações relevantes entre variáveis ambientais e os incêndios, evidenciando que esse problema é multifatorial e exige ações integradas.
- ▶ Portanto, é urgente que políticas públicas sejam fortalecidas, a fiscalização ambiental ampliada e a sociedade mobilizada para preservar nossas florestas. Mais do que números, os dados revelam uma ameaça concreta ao meio ambiente, à biodiversidade e à qualidade de vida das próximas gerações.