

```
import pandas as pd
bdf=pd.read_csv('/content/drive/MyDrive/Copy of brazil _forest_fire.csv',encoding='iso-8859-1')
```

```
bdf
```

	year	state	month	number	date
<b>0</b>	1998	Acre	Janeiro	0.0	1998-01-01
<b>1</b>	1999	Acre	Janeiro	0.0	1999-01-01
<b>2</b>	2000	Acre	Janeiro	0.0	2000-01-01
<b>3</b>	2001	Acre	Janeiro	0.0	2001-01-01
<b>4</b>	2002	Acre	Janeiro	0.0	2002-01-01
...	...	...	...	...	...
<b>6449</b>	2012	Tocantins	Dezembro	128.0	2012-01-01
<b>6450</b>	2013	Tocantins	Dezembro	85.0	2013-01-01
<b>6451</b>	2014	Tocantins	Dezembro	223.0	2014-01-01
<b>6452</b>	2015	Tocantins	Dezembro	373.0	2015-01-01
<b>6453</b>	2016	Tocantins	Dezembro	119.0	2016-01-01

6454 rows × 5 columns

## 1.Display Top5 rows?

```
bdf.head(5)
```

	year	state	month	number	date	month_eng
<b>0</b>	1998	Acre	Janeiro	0.0	1998-01-01	Jan
<b>1</b>	1999	Acre	Janeiro	0.0	1999-01-01	Jan
<b>2</b>	2000	Acre	Janeiro	0.0	2000-01-01	Jan
<b>3</b>	2001	Acre	Janeiro	0.0	2001-01-01	Jan
<b>4</b>	2002	Acre	Janeiro	0.0	2002-01-01	Jan

## 2. Display last five rows

```
bdf.tail(5)
```

	year	state	month	number	date	month_eng
6449	2012	Tocantins	Dezembro	128.0	2012-01-01	Dec

### 3. print No of rows and column

```
row,column=bdf.shape
print('No of Rows:',row)
print('No of Column:',column)
```

```
No of Rows: 6454
No of Column: 6
```

### 4. Get information about datasets

```
bdf.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6454 entries, 0 to 6453
Data columns (total 6 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
5   month_eng   6454 non-null   object
dtypes: float64(1), int64(1), object(4)
memory usage: 302.7+ KB
```

### 5. Check for duplicate data and drop. df.duplicated().any() newdf=df.drop\_duplicates()

```
bdf.duplicated().any()
```

```
True
```

```
newdf=bdf.drop_duplicates()
newdf
```

	year	state	month	number	date	month_eng
0	1998	Acre	Janeiro	0.0	1998-01-01	Jan
1	1999	Acre	Janeiro	0.0	1999-01-01	Jan

5. Check null values of Each columns

```
3    2001    Acre    Janeiro    0.0    2001-01-01    Jan
bdf.isnull().sum()
```

```
year      0
state     0
month     0
number    0
date      0
month_eng 0
dtype: int64
```

6452	2015	Tocantins	Dezembro	373.0	2015-01-01	Dec
------	------	-----------	----------	-------	------------	-----

7. Get overall statistics about dataframe?

```
6422 rows x 6 columns
bdf.describe()
```

	year	number
count	6454.000000	6454.000000
mean	2007.461729	108.293163
std	5.746654	190.812242
min	1998.000000	0.000000
25%	2002.000000	3.000000
50%	2007.000000	24.000000
75%	2012.000000	113.000000
max	2017.000000	998.000000

8. Rename Monthname to English.

```
bdf['month_eng']=bdf['month'].map({'Janeiro':'Jan',
'Fevereiro':'Feb',
'Março':'Mar',
'Abril':'Apr',
'Maio':'May',
'Junho':'Jun',
'Julho':'Jul',
'Agosto':'Aug',
'Setembro':'Sep',
'Outubro':'Oct',
'Novembro':'Nov',
'Dezembro':'Dec'})
bdf['month_eng']

0    Jan
1    Jan
2    Jan
3    Jan
```

```

4      Jan
      ...
6449   Dec
6450   Dec
6451   Dec
6452   Dec
6453   Dec
Name: month_eng, Length: 6454, dtype: object

```

### 9. In which month max no of forest fires were reported.

```

bdf.groupby('month_eng')['number'].sum().idxmax()

'Jul'

```

### 10. In which year No of forest fires were reported maximum.

```

bdf.groupby('year')['number'].sum().idxmax()

2003

```

### 11. In which state No of forest fires were reported.

```

bdf.groupby('state')['number'].sum().idxmax()

'Mato Grosso'

```

### 12. find total no of forest fires were reported in amozones

---

```

amazon=newdf.groupby('state')['number'].sum()
amazon['Amazonas']

30650.129

```

### 13. Display no of fires were reported in Amazonas(year-wise)

```

amz=newdf[newdf['state'].str.contains('Amazonas')]
fires_per_year = amz.groupby(['year'])['number'].sum()
fires_per_year

```

```

year
1998    946.000
1999   1061.000
2000    853.000
2001   1297.000
2002   2852.000
2003   1524.268
2004   2298.207
2005   1657.128
2006    997.640
2007    589.601
2008   2717.000
2009   1320.601
2010   2324.508
2011   1652.538

```

```

2012    1110.641
2013     905.217
2014    2385.909
2015    1189.994
2016    2060.972
2017     906.905
Name: number, dtype: float64

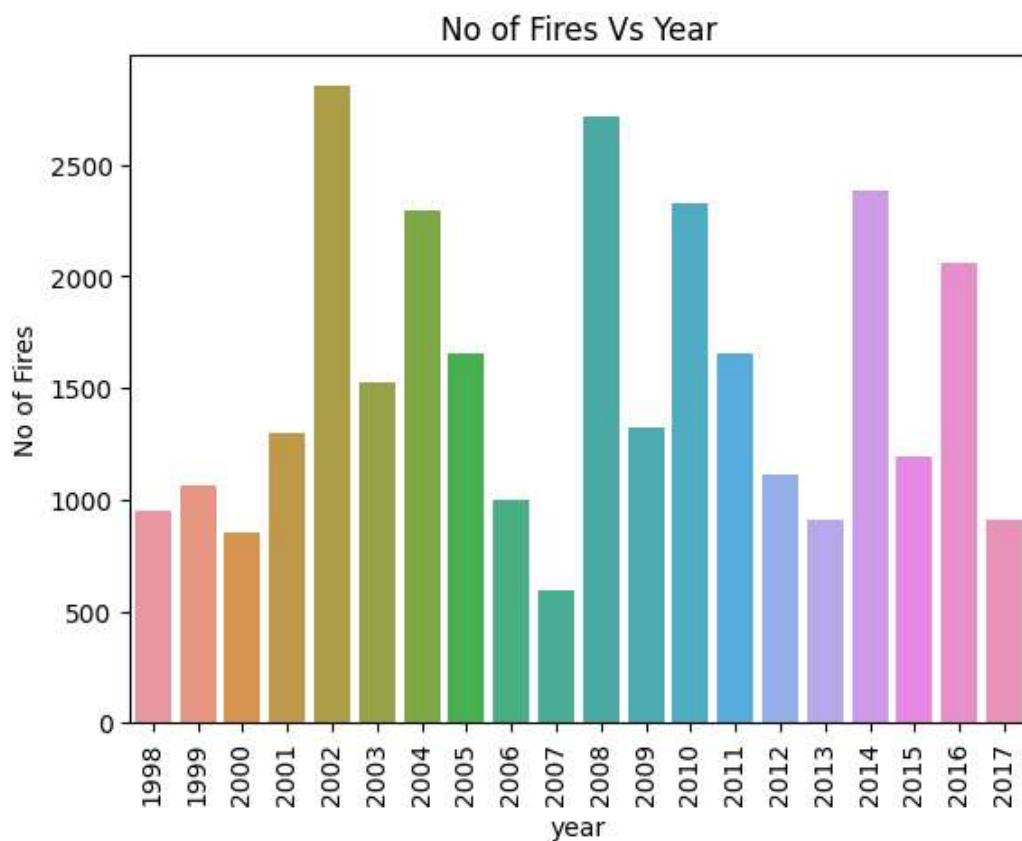
```

```

import seaborn as sb
import matplotlib.pyplot as plt
sb.barplot(x=fires_per_year.index,y=fires_per_year.values)
plt.xticks(rotation='vertical')
plt.title('No of Fires Vs Year')
plt.ylabel('No of Fires')

```

```
Text(0, 0.5, 'No of Fires')
```



#### 14. Display No fo fires were reported in amazonas(Month-wise)

```

amz=bdf[bdf['state'].str.contains('Amazonas')]
amz

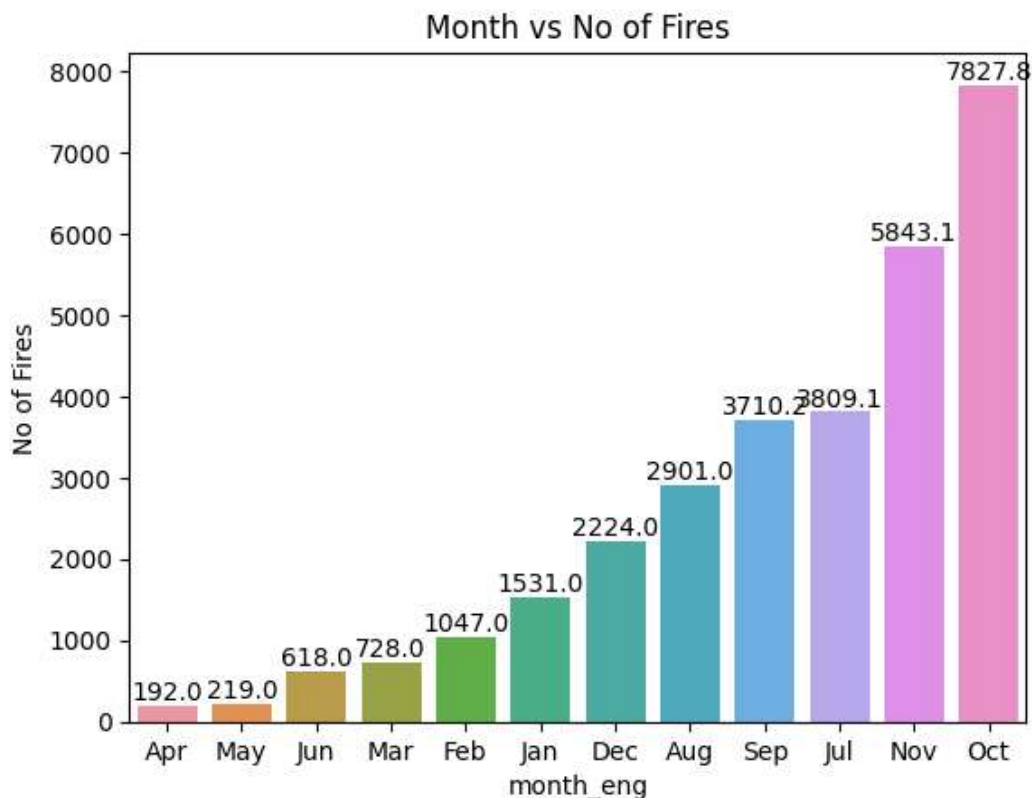
```

	year	state	month	number	date	month_eng
<b>718</b>	1998	Amazonas	Janeiro	0.0	1998-01-01	Jan
<b>719</b>	1999	Amazonas	Janeiro	3.0	1999-01-01	Jan
<b>720</b>	2000	Amazonas	Janeiro	7.0	2000-01-01	Jan
<b>721</b>	2001	Amazonas	Janeiro	3.0	2001-01-01	Jan
<b>722</b>	2002	Amazonas	Janeiro	17.0	2002-01-01	Jan

```
sr=amz.groupby('month_eng')['number'].sum()
data=sr.sort_values(ascending=True)
data
```

```
month_eng
Apr      192.000
May      219.000
Jun      618.000
Mar      728.000
Feb     1047.000
Jan     1531.000
Dec     2224.000
Aug     2900.974
Sep     3710.180
Jul     3809.112
Nov     5843.054
Oct     7827.809
Name: number, dtype: float64
```

```
ax=sb.barplot(x=data.index,y=data.values)
plt.title('Month vs No of Fires')
plt.ylabel('No of Fires')
for p in ax.patches:
    height = p.get_height()
    ax.text(p.get_x() + p.get_width() / 2., height, f'{height:.1f}', ha='center', va='bottom', font
```



**16. Find the total no of fires were reported in 2015 and visualized data based on each month?**

```
new=bdf[bdf['year']==2015]
new
```

	year	state	month	number	date	month_eng
<b>17</b>	2015	Acre	Janeiro	1.000	2015-01-01	Jan
<b>37</b>	2015	Acre	Fevereiro	2.000	2015-01-01	Feb
<b>57</b>	2015	Acre	Março	2.000	2015-01-01	Mar
<b>77</b>	2015	Acre	Abril	3.000	2015-01-01	Apr
<b>97</b>	2015	Acre	Maio	2.000	2015-01-01	May
...	...	...	...	...	...	...
<b>6372</b>	2015	Tocantins	Agosto	2.540	2015-01-01	Aug
<b>6392</b>	2015	Tocantins	Setembro	5.510	2015-01-01	Sep
<b>6412</b>	2015	Tocantins	Outubro	4.844	2015-01-01	Oct
<b>6432</b>	2015	Tocantins	Novembro	833.000	2015-01-01	Nov
<b>6452</b>	2015	Tocantins	Dezembro	373.000	2015-01-01	Dec

324 rows × 6 columns

```
m=new.groupby('month_eng')['number'].sum()
ax=sb.barplot(x=m.index,y=m.values)
for p in ax.patches:
    height = p.get_height()
    ax.text(p.get_x() + p.get_width() / 2., height, f'{height:.1f}', ha='center', va='bottom', font
plt.ylabel('No of Fires')
plt.title('No of Fires Vs Month')
```

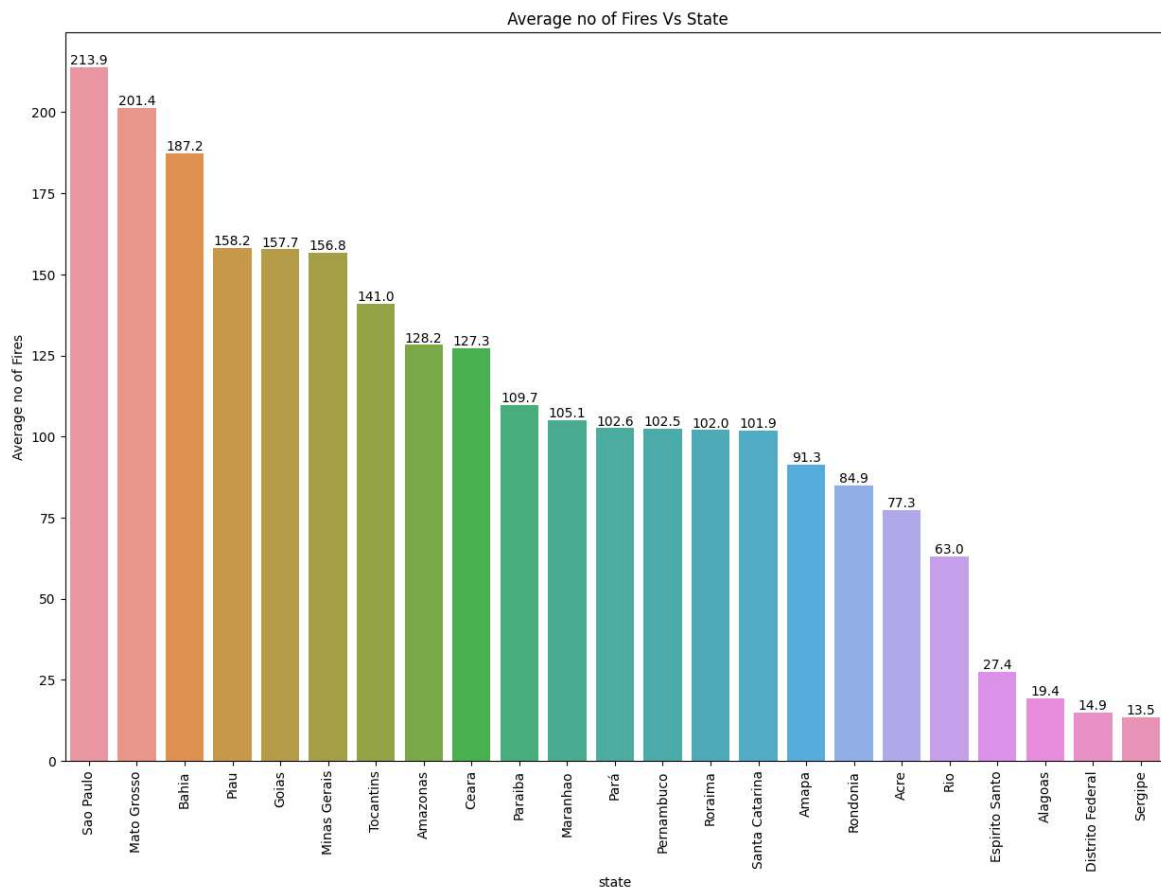
Text(0.5, 1.0, 'No of Fires Vs Month')

No of Fires Vs Month

## 17. Find Average no of fires reported from highest to lowest(state-wise) visualize the data.

```
s=bdf.groupby('state')['number'].mean()
x=s.sort_values(ascending=False)
plt.figure(figsize=(15, 10))
data=sb.barplot(x=x.index,y=x.values)

plt.title(' Average no of Fires Vs State')
plt.ylabel('Average no of Fires')
plt.xticks(rotation='vertical')
for i in data.patches:
    height=i.get_height()
    data.text(i.get_x() + i.get_width() / 2., height, f'{height:.1f}', ha='center', va='bottom', font
```



## 18. To find the state names where fires were reported in december month.



```
st=bdf[bdf['month_eng']=='Dec']  
s=st.groupby('state')['number'].sum()  
s  
plt.figure(figsize=(12,8))  
sb.barplot(x=s.index,y=s.values)  
plt.xticks(rotation=90)
```

```
ls=bdf['month'].unique()
list(ls)
```

```
bdf['month'].value_counts()
```

Al  
gc  
ma  
cor  
Bal  
Cea  
de  
Sal  
Go  
nh  
ro:  
ier  
rai  
Pã  
bu  
Pi  
I  
dor  
air  
ar  
Pa  
rgi  
nt

Highest No of Fires : '42760.674' in Year: '2003'

