```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.graph_objects as go
%matplotlib inline

from google.colab import files
uploaded=files.upload()
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving Brazil Deforestation.csv to Brazil Deforestation.csv

```
print(type(uploaded))
print(len(uploaded))
print(uploaded.keys())
```

```
<class 'dict'>
1
dict_keys(['Brazil Deforestation.csv'])
```

+ Code + Text

```
df=pd.read_csv('Brazil Deforestation.csv')
```

```
df.head()
```

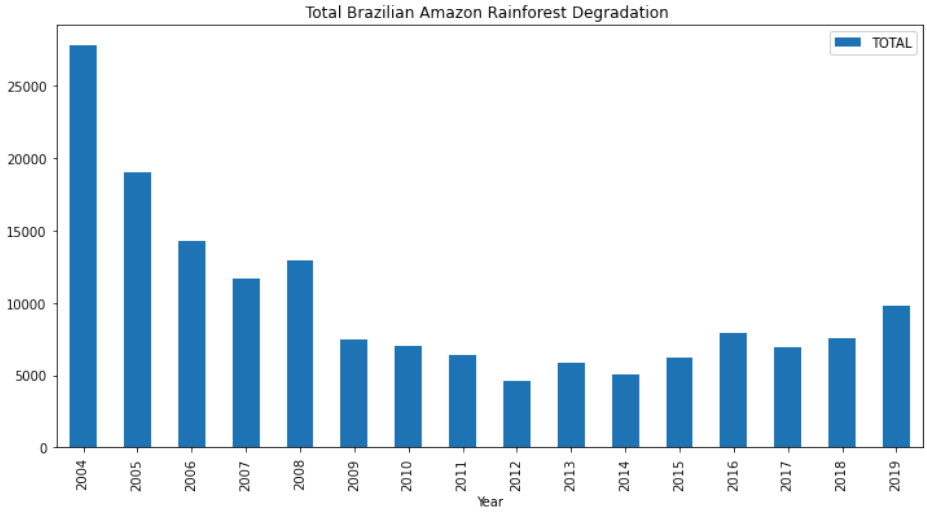
	Year	AC	AM	AP	MA	MT	PA	RO	RR	TO	TOTAL
0	2004	728	1232	46	755	11814	8870	3858	311	158	27772
1	2005	592	775	33	922	7145	5899	3244	133	271	19014
2	2006	398	788	30	674	4333	5659	2049	231	124	14286
3	2007	184	610	39	631	2678	5526	1611	309	63	11651
4	2008	254	604	100	1271	3258	5607	1136	574	107	12911

```
df.tail()
```

	Year	AC	AM	AP	MA	MT	PA	RO	RR	TO	TOTAL
11	2015	264	712	25	209	1601	2153	1030	156	57	6207
12	2016	372	1129	17	258	1489	2992	1376	202	58	7893
13	2017	257	1001	24	265	1561	2433	1243	132	31	6947
14	2018	444	1045	24	253	1490	2744	1316	195	25	7536
15	2019	688	1421	8	215	1685	3862	1245	617	21	9762

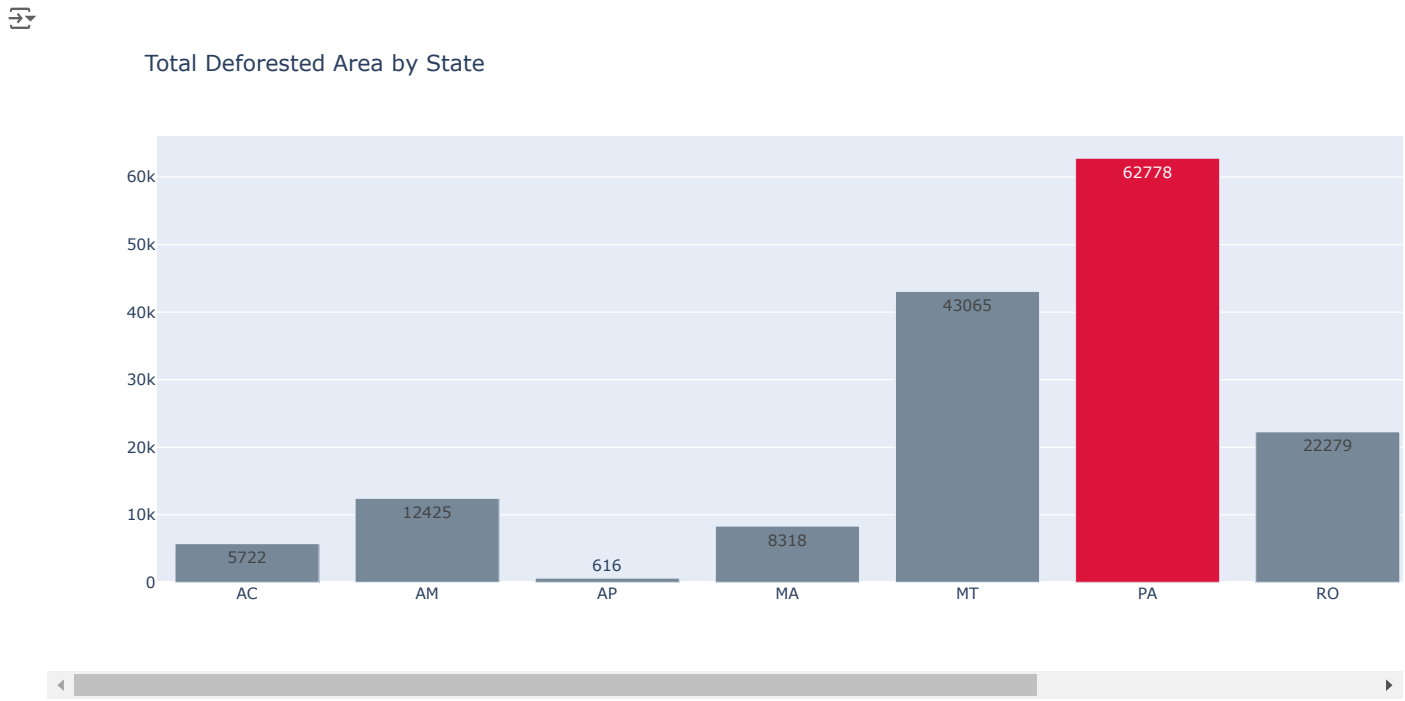
```
df.plot(kind='bar',x='Year',y='TOTAL',figsize=(12,6))
plt.title("Total Brazilian Amazon Rainforest Degradation ")
```

```
Text(0.5, 1.0, 'Total Brazilian Amazon Rainforest Degradation ')
```



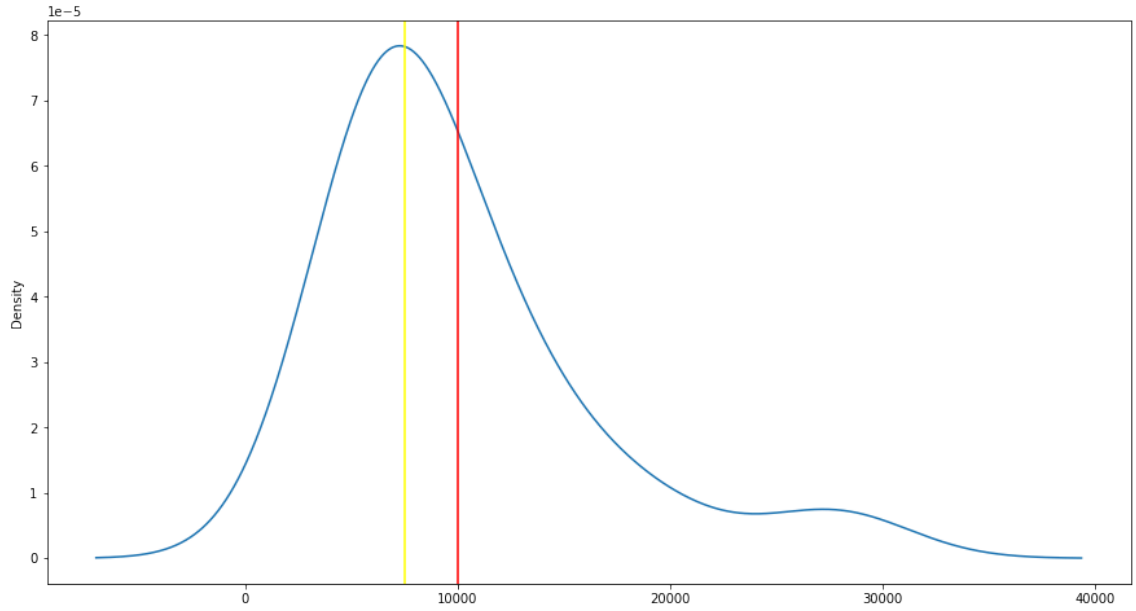

```
states=["AC","AM","AP","MA","MT","PA","RO","RR","TO"]
data=[]

for i in range(len(states)):
```

```
    data.append(df[states[i]].sum())

colors = ['lightslategray',] * 9
colors[5] = 'crimson'

fig = go.Figure(data=[go.Bar(x=states, y=data, text=data, textposition='auto', marker_color=colors)])
fig.update_layout(title_text='Total Deforested Area by State')

fig.show()
```

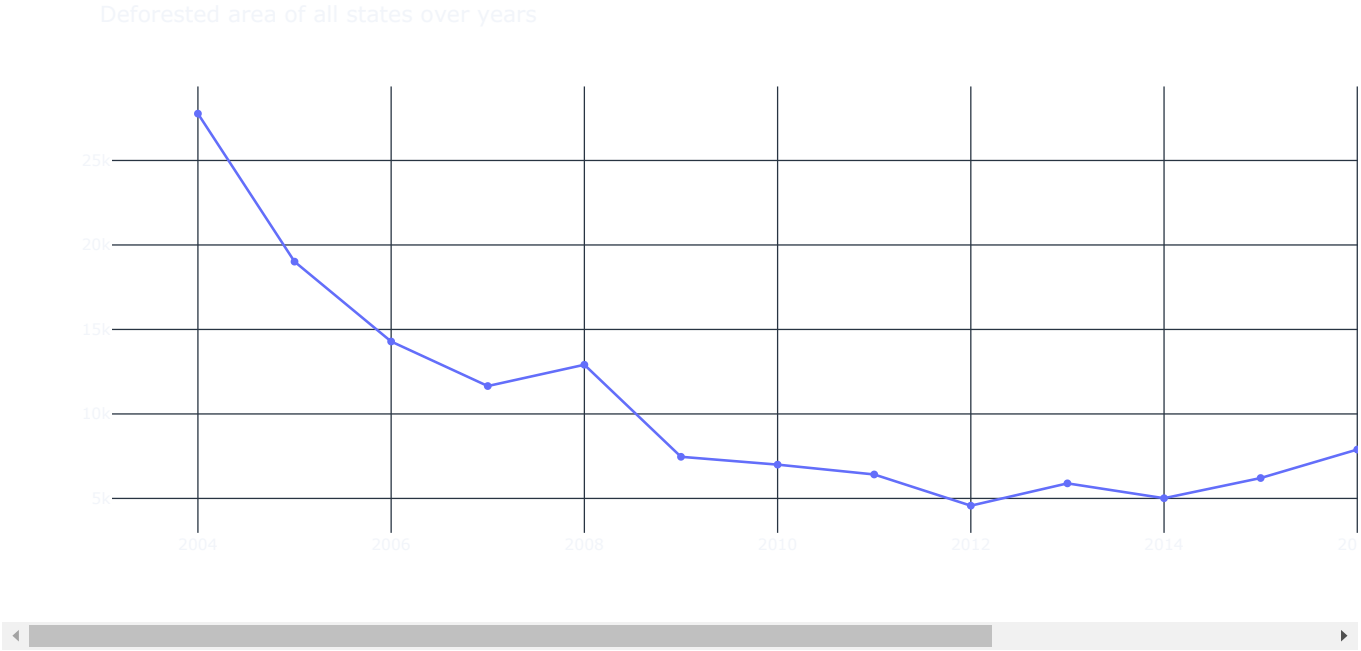



```
ax=df['TOTAL'].plot(kind='density',figsize=(15,8))
ax.axvline(df['TOTAL'].mean(),color='red')
ax.axvline(df['TOTAL'].median(),color='yellow')
```

```
<matplotlib.lines.Line2D at 0x7f3b4422b150>
```




```
trace=go.Scatter(x=df['Year'],y=df['TOTAL'])
layout=go.Layout(title="Deforested area of all states over years",template="plotly_dark")
deforest_p=[trace]
fig=go.Figure(deforest_p,layout=layout)
fig.show()
```




```
ax=df['TOTAL'].plot(kind='pie',figsize=(10,10))
mylabels = ['2004','2005','2006','2007','2008','2009','2010','2011','2012','2013','2014','2015','2016','2017','2018','2019']

plt.pie(ax, labels = mylabels)
plt.show()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-14-8c98596e4be7> in <module>
      2 mylabels = ['2004','2005','2006','2007','2008','2009','2010','2011','2012','2013','2014','2015','2016','2017','2018','2019']
      3
----> 4 plt.pie(ax, labels = mylabels)
      5 plt.show()

                                  2 frames
/usr/local/lib/python3.7/dist-packages/matplotlib/axes/_axes.py in pie(self, x, explode, labels, colors, autopct, pctdistance, shadow, labeldistance, startangle, radius, counterclock, wedgeprops, textprops, center, frame, rotatelabels)
   2910         # The use of float32 is "historical", but can't be changed without
   2911         # regenerating the test baselines.
-> 2912         x = np.asarray(x, np.float32)
   2913         if x.ndim != 1 and x.squeeze().ndim <= 1:
   2914             cbook.warn_deprecated(

TypeError: float() argument must be a string or a number, not 'AxesSubplot'
```