P4DS (ODL1) Assignment 3

Data Analysis Project

Notebook template design: Brandon Bennett (2020.11.03, revised 2021.03.02)

Brazilian Amazon Rainforest Degradation

Student: Siddharth Sahu

Email: od21ss@leeds.ac.uk

Project Plan

The Data (10 marks)

This report analyzes a dataset pertaining to the degradation of the Amazon rainforest in Brazil. The dataset is downloaded from Kaggle website and consists of two CSV files. One of the files contains data on the deforestation area(km^2) of Brazil's Legal Amazon (BLA) states, also known as *Amazônia Legal* between the years 2004 and 2019. And the other file contains the data on fire outbreaks for the same states between the years 1999 and 2019.

The Kaggle user had acquired the data from the websites of the monitoring programs PRODES and BDQ, administered by INPE(*Instituto Nacional de Pesquisas Espaciais* also known as the National Institute for Space Research in English), which is a research unit of the Brazilian Ministry of Science, Technology and Innovations. As per the user, data was extracted as-is from the INPE program websites in December 2019. Since the data was last updated two years ago, there is no data for 2020 and 2021 in the dataset.

The CSV file **_def_area_2004*2019.csv*** holds the deforestation area data in km^2 for each BLA state and, has 16 data rows, 1 header row and 11 columns. The dataset in this CSV file is addressed as "*deforestation-dataset*" in this report. The details of the columns in the deforestation-dataset are tabulated below:

Column	Datatype	Description
Ano/Estados	int64	The column name is in Portuguese and contains the year of occurrence
AC	int64	Contains deforested area (km^2) in Acre state
AM	int64	Contains deforested area (km^2) in Amazonas state
AP	int64	Contains deforested area (km^2) in Amapa state
MA	int64	Contains deforested area (km^2) in Maranhao state
MT	int64	Contains deforested area (km^2) in Mato Grosso state
PA	int64	Contains deforested area (km^2) in Para state
RO	int64	Contains deforested area (km^2) in Rondônia state
RR	int64	Contains deforested area (km^2) in Roraima state
TO	int64	Contains deforested area (km^2) in Tocantins state
AMZ LEGAL	int64	Contains the sum of deforested area(km^2) in Amazonia Legal

The CSV file **_inpe_brazilian_amazon_fires_1999*2019.csv*** holds the number of forest fire outbreaks in each BLA state and has 2104 data rows, 1 header row and 6 columns. The dataset in this CSV file is addressed as "*firespots-dataset*" in this report. The details of the columns in the firespots-dataset are tabulated below:

Column	Datatype	Description
year	int64	Contains the year of occurrence
month	int64	Contains the month of occurrence
state	object	State of Amazonia Legal
latitude	float64	Latitude of fire outbreaks
longitude	float64	Longitude of fire outbreaks
firespots	int64	Number of forest fire outbreaks

The data in both of the files have no missing values or NaN for the numeric data types. In addition, data is significantly cleaned as most of the columns are numeric types except for the states in the firespots-dataset, which is an object type. Since the state column contains the name of the states, so no data conversion is required. However, both the files are unaligned in terms of the year of data stored in each file. In the firespots-dataset, the year column is ranged from 1999 to 2019, whereas in the deforestation-dataset, it is ranged from 2004 to 2019.

Project Aim and Objectives (5 marks)

Amazon rainforest, also known as *Amazonia*, is the largest and most biodiverse tropical rainforest covering approx. 5,500,000 km^2 of land area. Amazonia is home to different species of trees, insects, birds, and mammals. Amazonia is losing its biodiversity because of deforestation and forest fires. This will result in global warming due to the amount of carbon trapped by the vegetation released into the atmosphere. Brazil plays a significant role in conserving the rainforest as it contains 60% of the rainforest.

Deforestation is the process of converting forested land into non-forested land. Some of the primary reasons for deforestation are illegal logging, agricultural expansion, urbanization, and even fire outbreaks. To analyze hidden pattern and trend of deforestation and fire outbreaks, I will explore and visualize the historical data obtained.

Specific Objective(s)

- **Objective 1:** *To analyze the trend of the deforestation in the states of the Amazon rainforest in Brazil.*
- **Objective 2:** *To analyze the occurance of fire outbreaks in the states of the Amazon rainforest in Brazil.*
- **Objective 3:** *To correlate the deforestation data and the number of fire outbreaks in the states of the Amazon rainforest in Brazil.*

System Design (5 marks)

Architecture

The overall architecture of this project is shown in the flow diagram below. The implementation process starts with the importing of the libraries and packages required for carrying out data preparation and data visualisation. Then the CSV files of the datasets collected from Kaggle are loaded into the Panda dataframes. The next step in the process pipeline is preparation of the data, in which the overall structure of the datasets are investigated, and the dataframes are modified and transformed to prepare them for exploration and analysis.

In the data exploration step, functions are developed to study the distribution of the two datasets - deforestation-dataset and firespots-dataset - over all the years for the nine BLA states. The exploration step provides a better understanding of the datasets and is thus cruicial to the whole process of data analysis. The next step is to analyse the data for the predefined objectives. Various trends in the deforestation and firespots are analysed along with the study of correlations between the two. State-wise and year-wise data were analysed for both the datsets and month-wise data was analysed only for the firespots-dataset. All the trends and correlations found are visualized using bar plots, line plots, heatmaps and pie-charts.

Processing Modules and Algorithms

The following modules are implemented in this project:

1. Data import

> - reading of the CSV files from the local directory

2. Data preparation

> - removing the invalid entries if available
> - renaming of the column names
> - creating the new data structures

3. Data Exploration

> - searching in the data
> - transforming the dataframes by appending feature vectors

4. Data Analysis and Visualization

> - analyzing the data
> - plotting the graphs
> - generating coorelations between the two datasets used

Program Code (15 marks)

Data Import

Importing the required python libraries

In [1]:

```
import pandas as pd                 # Importing pandas for data analysis and manipulation
import matplotlib.pyplot as plt     # Importing matplotlib.pyplot for plotting graphs
import seaborn as sns               # Importing seaborn for visualizing data
```

Reading the CSV files

Store the CSV files in the notebook working directory so that files will be read by pandas read_csv() function.

In [2]:
```
BRZ_AMZ_DEFOREST_DF = pd.read_csv("def_area_2004_2019.csv")
BRZ_AMZ_FIREOUTBREAK_DF = pd.read_csv("inpe_brazilian_amazon_fires_1999_2019.csv")
```

Above cell read the CSV files from the working directory and and load the data into dataframes.

Data Preparation

Understanding the data

To get information about the columns, call info() and describe() on pandas dataframe.

In [3]:
```
BRZ_AMZ_DEFOREST_DF.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16 entries, 0 to 15
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Ano/Estados  16 non-null     int64
 1   AC           16 non-null     int64
 2   AM           16 non-null     int64
 3   AP           16 non-null     int64
 4   MA           16 non-null     int64
 5   MT           16 non-null     int64
 6   PA           16 non-null     int64
 7   RO           16 non-null     int64
 8   RR           16 non-null     int64
 9   TO           16 non-null     int64
 10  AMZ LEGAL    16 non-null     int64
dtypes: int64(11)
memory usage: 1.5 KB
```

Output of above cell shows information related to _BRZ_AMZ_DEFOREST*DF* dataframe. There are 16 entries i.e. rows in the dataframe indexed from 0 to 15 and 11 columns which have no null values.

In [4]:
```
BRZ_AMZ_DEFOREST_DF.describe()
```

Out[4]:

	Ano/Estados	AC	AM	AP	MA	MT	PA	RO	RR	TO	AMZ LEGAL
count	16.000000	16.000000	16.000000	16.000000	16.000000	16.000000	16.000000	16.000000	16.000000	16.000000	16.000000
mean	2011.500000	357.625000	776.562500	38.500000	519.875000	2691.562500	3923.625000	1392.437500	243.187500	77.562500	10020.937500
std	4.760952	172.476037	300.648625	23.557023	316.705305	2940.367595	1945.802487	943.567413	150.567468	63.238141	6112.467134
min	2004.000000	167.000000	405.000000	8.000000	209.000000	757.000000	1741.000000	435.000000	121.000000	21.000000	4571.000000
25%	2007.750000	256.250000	568.000000	24.000000	257.750000	1108.750000	2411.250000	842.000000	139.000000	46.750000	6365.250000
50%	2011.500000	292.500000	661.000000	30.500000	399.500000	1525.500000	3389.000000	1189.500000	198.500000	57.500000	7500.000000
75%	2015.250000	409.500000	1012.000000	47.750000	722.750000	2823.000000	5546.250000	1434.750000	269.250000	82.250000	11966.000000
max	2019.000000	728.000000	1421.000000	100.000000	1271.000000	11814.000000	8870.000000	3858.000000	617.000000	271.000000	27772.000000

Output of above cell shows statistical data related to each columns in the _BRZ_AMZ_DEFOREST*DF* dataframe.

In [5]:
```
BRZ_AMZ_FIREOUTBREAK_DF.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2104 entries, 0 to 2103
Data columns (total 6 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   year       2104 non-null   int64
 1   month      2104 non-null   int64
 2   state      2104 non-null   object
 3   latitude   2104 non-null   float64
 4   longitude  2104 non-null   float64
 5   firespots  2104 non-null   int64
dtypes: float64(2), int64(3), object(1)
memory usage: 98.8+ KB
```

Output of above cell shows information related to _BRZ_AMZ_FIREOUTBREAK*DF* dataframe. There are 2104 entries i.e. rows in the dataframe indexed from 0 to 2103 and 6 columns which have no null values.

In [6]:
```
BRZ_AMZ_FIREOUTBREAK_DF.describe()
```

Out[6]:

	year	month	latitude	longitude	firespots
count	2104.000000	2104.000000	2104.000000	2104.000000	2104.000000
mean	2009.267110	6.704848	-5.439282	-56.510314	1167.417776

	year	month	latitude	longitude	firespots
std	5.963029	3.431949	4.852439	7.382619	2959.558714
min	1999.000000	1.000000	-14.431908	-73.085000	1.000000
25%	2004.000000	4.000000	-9.946974	-62.376004	16.000000
50%	2009.000000	7.000000	-5.808292	-55.876480	121.000000
75%	2014.000000	10.000000	-2.595169	-50.396154	755.000000
max	2019.000000	12.000000	4.151000	-44.188000	37926.000000

Output of above cell shows statistical data related to each columns in the _BRZ_AMZ_FIREOUTBREAK*DF* dataframe.

A Glance of the dataset

To peek into the dataset, call head() method on pandas dataframe.

In [7]:

```
BRZ_AMZ_DEFOREST_DF.head(10)
```

Out[7]:

	Ano/Estados	AC	AM	AP	MA	MT	PA	RO	RR	TO	AMZ LEGAL
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
5	2009	167	405	70	828	1049	4281	482	121	61	7464
6	2010	259	595	53	712	871	3770	435	256	49	7000
7	2011	280	502	66	396	1120	3008	865	141	40	6418
8	2012	305	523	27	269	757	1741	773	124	52	4571
9	2013	221	583	23	403	1139	2346	932	170	74	5891

Output of above cell shows top ten rows of deforestation-dataset.

In [8]:

```
BRZ_AMZ_FIREOUTBREAK_DF.head(10)
```

Out[8]:

	year	month	state	latitude	longitude	firespots
0	1999	1	AMAZONAS	-2.371113	-59.899933	3
1	1999	1	MARANHAO	-2.257395	-45.487831	36
2	1999	1	MATO GROSSO	-12.660633	-55.057989	18
3	1999	1	PARA	-2.474820	-48.546967	87
4	1999	1	RONDONIA	-12.861700	-60.513100	1
5	1999	1	RORAIMA	3.403225	-60.622853	15
6	1999	2	AMAPA	-0.155000	-52.683100	1
7	1999	2	AMAZONAS	-2.763167	-63.429781	43
8	1999	2	MATO GROSSO	-12.619988	-55.375363	8
9	1999	2	PARA	-2.150617	-53.509911	285

Output of above cell shows top ten rows of firespots-dataset.

Renaming the columns

Rename the columns of deforestation dataset to meaningful column names in english and assigning the data to new dataframe.

In [9]:

```
DEFOREST_DF = BRZ_AMZ_DEFOREST_DF.rename(columns={'Ano/Estados':'year',
                                                   'AMZ LEGAL':'total_deforestation'})

DEFOREST_DF.head()
```

Out[9]:

	year	AC	AM	AP	MA	MT	PA	RO	RR	TO	total_deforestation
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

Columns "*Ano/Estados*" and "*AMZ LEGAL*" are changed to "year" and "total_deforestation" and assigned all the data to new dataframe.

Creating a list of states

Create a list of shortnames of the BLA states provided in the deforestation-dataset for referring to the states in this project report.

In [10]:

```
STATES_SHORTNAME_LIST = BRZ_AMZ_DEFOREST_DF.columns[1:10]
print(STATES_SHORTNAME_LIST)
```

```
Index(['AC', 'AM', 'AP', 'MA', 'MT', 'PA', 'RO', 'RR', 'TO'], dtype='object')
```

Building dictionaries

Create a dictionary to store shortnames of the states as keys and fullnames of the states as values. This dictionary will be used to get fullname of a state provided its shortname.

In [11]:

```
STATES_FULLNAME_DICT = dict.fromkeys(STATES_SHORTNAME_LIST, 0)      # Create empty dictionary with keys from the list

STATES_FULLNAME_DICT['AC'] = 'ACRE'
STATES_FULLNAME_DICT['AM'] = 'AMAZONAS'
STATES_FULLNAME_DICT['AP'] = 'AMAPA'
STATES_FULLNAME_DICT['MA'] = 'MARANHAO'
STATES_FULLNAME_DICT['MT'] = 'MATO GROSSO'
STATES_FULLNAME_DICT['PA'] = 'PARA'
STATES_FULLNAME_DICT['RO'] = 'RONDONIA'
STATES_FULLNAME_DICT['RR'] = 'RORAIMA'
STATES_FULLNAME_DICT['TO'] = 'TOCANTINS'

print(STATES_FULLNAME_DICT)
```

```
{'AC': 'ACRE', 'AM': 'AMAZONAS', 'AP': 'AMAPA', 'MA': 'MARANHAO', 'MT': 'MATO GROSSO', 'PA': 'PARA', 'RO': 'RONDONIA', 'RR': 'RORA
IMA', 'TO': 'TOCANTINS'}
```

Creating a dictionary to store the numeric values of the months as keys and the month names as values.

In [12]:

```
MONTH_DICT = {
    1:'JANUARY',
    2:'FEBRUARY',
    3:'MARCH',
    4:'APRIL',
    5:'MAY',
    6:'JUNE',
    7:'JULY',
    8:'AUGUST',
    9:'SEPTEMBER',
    10:'OCTOBER',
    11:'NOVEMBER',
    12:'DECEMBER'
}

print(MONTH_DICT)
```

```
{1: 'JANUARY', 2: 'FEBRUARY', 3: 'MARCH', 4: 'APRIL', 5: 'MAY', 6: 'JUNE', 7: 'JULY', 8: 'AUGUST', 9: 'SEPTEMBER', 10: 'OCTOBER',
11: 'NOVEMBER', 12: 'DECEMBER'}
```

Data Exploration

Searching in the datasets

Function to return the deforestation area given a state and a year

In [13]:

```
def get_deforest_area(year, state):
    if state not in DEFOREST_DF.columns:                    # Check if column_name is present in dataframe
        print("State '{}' is not found in the dataframe".format(state))
        return None

    return int(DEFOREST_DF[(DEFOREST_DF['year'] == year)][state])

get_deforest_area(2009, 'AC')
```

Out[13]:

```
167
```

Function to return the year of deforestation given a state and deforested area

In [14]:

```
def get_deforest_year(state, deforested_area):
    if state not in DEFOREST_DF.columns:                    # Check if column_name is present in dataframe
        print("State '{}' is not found in the dataframe".format(state))
        return None

    return int(DEFOREST_DF[DEFOREST_DF[state]==deforested_area]['year'])

get_deforest_year('AC', 167)
```

Out[14]:

```
2009
```

Function to return the state of deforestation provided a deforested area and a year

In [15]:

```
def get_deforest_state(deforested_area, year):
    DF = DEFOREST_DF[DEFOREST_DF['year']==year].iloc[:,1:10]            # Get the data of each states in the given year
    state = DF[DF==deforested_area].idxmax(axis=1).to_string(index=False)   # Extract the name of state from the column name
    return state

get_deforest_state(167, 2009)
```

Out[15]:

```
'AC'
```

Function to return the deforested area of each states provided a year

In [16]:

```
def get_deforest_area_state(year):
   return DEFOREST_DF[DEFOREST_DF['year']==year].iloc[:,1:10]

get_deforest_area_state(2009)
```

Out[16]:

	AC	AM	AP	MA	MT	PA	RO	RR	TO
5	167	405	70	828	1049	4281	482	121	61

Function to return the firespots provided a year, a month and a state from the firespots-dataset

In [17]:

```
def get_fire_outbreak_data(year, month, state):
    if state not in BRZ_AMZ_FIREOUTBREAK_DF['state'].unique():      # Check if state passed as argument is in the dataframe
        print("State '{}' is not found in the dataframe".format(state))
        return None

    fire_outbreaks = BRZ_AMZ_FIREOUTBREAK_DF[(BRZ_AMZ_FIREOUTBREAK_DF['year'] == year) &
                                             (BRZ_AMZ_FIREOUTBREAK_DF['month'] == month) &
                                             (BRZ_AMZ_FIREOUTBREAK_DF['state'] == state)]['firespots']
    return int(fire_outbreaks)

get_fire_outbreak_data(2004, 2, 'PARA')
```

Out[17]:

```
36
```

Function to return the minimum and the maximum deforested areas and corresponding years provided a state

In [18]:

```
def get_min_max_deforest_area_year(state):
    min_data = DEFOREST_DF[state].min()                           # Get min deforested area of the state
    min_data_year = get_deforest_year(state, min_data)    # Get the deforested year of min area of the state

    max_data = DEFOREST_DF[state].max()                           # Get min deforested area of the state
    max_data_year = get_deforest_year(state, max_data)    # Get the deforested year of max area of the state

    return {'min':(min_data_year, min_data),            # Return dictionary having min and max as keys and tuple of the year
            'max':(max_data_year, max_data)}            # and the area from dataframes as values.


get_min_max_deforest_area_year('AC')
```

Out[18]:

```
{'max': (2004, 728), 'min': (2009, 167)}
```

Function to return the minimum and the maximum deforested areas and corresponding state provided a year

In [19]:

```
def get_min_max_deforest_area_state(year):
    min_data_state = DEFOREST_DF[DEFOREST_DF['year']==year].iloc[:,1:10].idxmin(axis=1).to_string(index=False)
    min_data = int(DEFOREST_DF[DEFOREST_DF['year']==year][min_data_state])

    max_data_state = DEFOREST_DF[DEFOREST_DF['year']==year].iloc[:,1:10].idxmax(axis=1).to_string(index=False)
    max_data = int(DEFOREST_DF[DEFOREST_DF['year']==year][max_data_state])

    return {'min':(min_data_state, min_data),            # Return dictionary having min and max as keys and tuple of the year
            'max':(max_data_state, max_data)}            # and the area from dataframes as values.


get_min_max_deforest_area_state(2004)
```

Out[19]:

```
{'max': ('MT', 11814), 'min': ('AP', 46)}
```

Transforming the datasets

Function to return a restructured dataframe with yearly aggregated sum of firespots for each state and in each year and their sum total under **total_firespots** column

In [20]:

```
def agg_firespots_sum_yearly_per_state():
    NEW_DF = BRZ_AMZ_FIREOUTBREAK_DF[['year', 'state', 'firespots']]

    NEW_DF = NEW_DF.groupby(by=['year','state']).sum()
    NEW_DF.reset_index(inplace=True)
```

```
    NEW_DF = NEW_DF.pivot(index='year', columns='state', values='firespots')
    NEW_DF.reset_index(inplace=True)

    NEW_DF = NEW_DF.rename_axis(None, axis=1)
    NEW_DF['total_firespots'] = NEW_DF.iloc[:,1:10].sum(axis=1)

    return NEW_DF

agg_firespots_sum_yearly_per_state()
```

Out[20]:

	year	ACRE	AMAPA	AMAZONAS	MARANHAO	MATO GROSSO	PARA	RONDONIA	RORAIMA	TOCANTINS	total_firespots
0	1999	347	101	1048	4136	28538	20478	7121	220	869	62858
1	2000	430	253	857	4500	17242	18201	5505	362	818	48168
2	2001	829	1300	1297	7979	20795	28590	5062	2415	1408	69675
3	2002	7985	3730	10203	20848	79680	106849	39132	2845	2601	273873
4	2003	10523	2516	10191	12036	50713	53040	30533	3987	861	174400
5	2004	7271	3413	8083	11443	70422	74214	40824	2221	746	218637
6	2005	15993	2020	15644	10820	53489	71477	41641	1461	1175	213720
7	2006	6198	1665	11697	7885	32745	55840	25699	2189	504	144422
8	2007	8549	1484	11293	12329	52399	68491	27477	3244	1214	186480
9	2008	5699	2153	6701	7822	18602	48449	11549	1950	528	103453
10	2009	3511	2456	9280	6664	9970	41664	5396	2359	382	81682
11	2010	8661	1000	12139	7298	28362	57196	16924	1918	1116	134614
12	2011	3191	1396	5028	4976	9103	26563	6078	1475	376	58186
13	2012	4720	2518	9114	6919	16133	37221	8312	1376	406	86719
14	2013	4980	1529	6512	4418	10830	24046	4613	1395	365	58688
15	2014	4398	1848	9008	5733	15677	35526	7327	2519	518	82554
16	2015	5779	2936	13419	7475	17599	43164	13105	2452	509	106438
17	2016	7684	2595	11173	4928	15836	29724	11462	3870	489	87761
18	2017	6295	1946	11685	5896	18143	49770	11298	1565	841	107439
19	2018	6626	1206	11446	2449	11621	22080	10253	2383	281	68345
20	2019	6802	1272	12665	3989	17479	29700	11206	4775	247	88135

Function to return a restructured dataframe with monthly aggregated sum of firespots for each state and in each month and their sum total and yearly averages under **total_firespots** and **yearly_avg_firespots** columns, respectively

In [21]:

```
def agg_firespots_sum_monthly_per_state():
    NEW_DF = BRZ_AMZ_FIREOUTBREAK_DF[['month', 'state', 'firespots']]
    NEW_DF = NEW_DF.groupby(by=['month','state']).sum()
    NEW_DF.reset_index(inplace=True)

    NEW_DF = NEW_DF.pivot(index='month', columns='state', values='firespots')
    NEW_DF.reset_index(inplace=True)
    NEW_DF = NEW_DF.rename_axis(None, axis=1)

    NEW_DF['total_firespots'] = NEW_DF.iloc[:,1:10].sum(axis=1)
    NEW_DF['yearly_avg_firespots'] = NEW_DF['total_firespots']/(BRZ_AMZ_FIREOUTBREAK_DF['year'].max()-BRZ_AMZ_FIREOUTBREAK_DF['yea

    NEW_DF['month'] = NEW_DF['month'].apply(lambda x:MONTH_DICT[x])

    return NEW_DF

agg_firespots_sum_monthly_per_state()
```

Out[21]:

	month	ACRE	AMAPA	AMAZONAS	MARANHAO	MATO GROSSO	PARA	RONDONIA	RORAIMA	TOCANTINS	total_firespots	yearly_avg_firespots
0	JANUARY	63	458	2284	4108	4635	13613	579	8078	87	33905	1695.25
1	FEBRUARY	33	25	1282	344	4463	1823	299	8457	71	16797	839.85
2	MARCH	28	21	771	85	6518	757	328	13497	42	22047	1102.35
3	APRIL	116	21	333	80	7878	512	352	4607	31	13930	696.50
4	MAY	312	22	398	298	17417	1439	826	542	77	21331	1066.55
5	JUNE	640	44	1391	1030	44853	7053	3003	128	246	58388	2919.40
6	JULY	4712	72	13010	2961	56567	45978	14644	108	668	138720	6936.00
7	AUGUST	36083	402	63558	7049	139145	241683	93012	316	2686	583934	29196.70
8	SEPTEMBER	70313	2741	59857	14570	206432	208204	162965	1019	7527	733628	36681.40
9	OCTOBER	12871	11143	28676	35117	82499	148781	53842	2371	3620	378920	18946.00

	month	ACRE	AMAPA	AMAZONAS	MARANHAO	MATO GROSSO	PARA	RONDONIA	RORAIMA	TOCANTINS	total_firespots	yearly_avg_firespots
10	NOVEMBER	1217	17774	12770	57652	17490	183887	8633	3625	912	303960	15198.00
11	DECEMBER	83	6614	4153	37249	7481	88553	2034	4233	287	150687	7534.35

Data Analysis and Visualization

Analyzing the deforestation-dataset

Total deforestation in each year

Check the total deforestation data of each years from deforestation-dataset

In [22]:

```
DEFOREST_DF[['year', 'total_deforestation']]     # Give the data in the columns provided as list
```

Out[22]:

	year	total_deforestation
0	2004	27772
1	2005	19014
2	2006	14286
3	2007	11651
4	2008	12911
5	2009	7464
6	2010	7000
7	2011	6418
8	2012	4571
9	2013	5891
10	2014	5012
11	2015	6207
12	2016	7893
13	2017	6947
14	2018	7536
15	2019	9762

Above cell shows deforestation reduced over the years but started increasing from 2017.

Check the deforestation data of each state and each year

In [23]:

```
DEFOREST_DF.iloc[:,:-1]
```

Out[23]:

	year	AC	AM	AP	MA	MT	PA	RO	RR	TO
0	2004	728	1232	46	755	11814	8870	3858	311	158
1	2005	592	775	33	922	7145	5899	3244	133	271
2	2006	398	788	30	674	4333	5659	2049	231	124
3	2007	184	610	39	631	2678	5526	1611	309	63
4	2008	254	604	100	1271	3258	5607	1136	574	107
5	2009	167	405	70	828	1049	4281	482	121	61
6	2010	259	595	53	712	871	3770	435	256	49
7	2011	280	502	66	396	1120	3008	865	141	40
8	2012	305	523	27	269	757	1741	773	124	52
9	2013	221	583	23	403	1139	2346	932	170	74
10	2014	309	500	31	257	1075	1887	684	219	50
11	2015	264	712	25	209	1601	2153	1030	156	57
12	2016	372	1129	17	258	1489	2992	1376	202	58
13	2017	257	1001	24	265	1561	2433	1243	132	31
14	2018	444	1045	24	253	1490	2744	1316	195	25
15	2019	688	1421	8	215	1685	3862	1245	617	21

Above cell shows that state 'MT' has maximum deforested area in the year 2004.

Check the min and max deforestation for each year

In [24]:

```
for year in DEFOREST_DF['year']:
    print(year, '=', get_min_max_deforest_area_state(year))
```

```
2004 = {'min': ('AP', 46), 'max': ('MT', 11814)}
2005 = {'min': ('AP', 33), 'max': ('MT', 7145)}
2006 = {'min': ('AP', 30), 'max': ('PA', 5659)}
2007 = {'min': ('AP', 39), 'max': ('PA', 5526)}
2008 = {'min': ('AP', 100), 'max': ('PA', 5607)}
2009 = {'min': ('TO', 61), 'max': ('PA', 4281)}
2010 = {'min': ('TO', 49), 'max': ('PA', 3770)}
2011 = {'min': ('TO', 40), 'max': ('PA', 3008)}
2012 = {'min': ('AP', 27), 'max': ('PA', 1741)}
2013 = {'min': ('AP', 23), 'max': ('PA', 2346)}
2014 = {'min': ('AP', 31), 'max': ('PA', 1887)}
2015 = {'min': ('AP', 25), 'max': ('PA', 2153)}
2016 = {'min': ('AP', 17), 'max': ('PA', 2992)}
2017 = {'min': ('AP', 24), 'max': ('PA', 2433)}
2018 = {'min': ('AP', 24), 'max': ('PA', 2744)}
2019 = {'min': ('AP', 8), 'max': ('PA', 3862)}
```

Above cell shows that in 2004 and 2005, 'MT' has maximum deforestation, whereas after 2005, 'PA' has maximum deforested area in each years.

Check the min and max deforestation for each store

In [25]:

```
for state in STATES_SHORTNAME_LIST:
    print(STATES_FULLNAME_DICT[state], '=', get_min_max_deforest_area_year(state))
```

```
ACRE = {'min': (2009, 167), 'max': (2004, 728)}
AMAZONAS = {'min': (2009, 405), 'max': (2019, 1421)}
AMAPA = {'min': (2019, 8), 'max': (2008, 100)}
MARANHAO = {'min': (2015, 209), 'max': (2008, 1271)}
MATO GROSSO = {'min': (2012, 757), 'max': (2004, 11814)}
PARA = {'min': (2012, 1741), 'max': (2004, 8870)}
RONDONIA = {'min': (2010, 435), 'max': (2004, 3858)}
RORAIMA = {'min': (2009, 121), 'max': (2019, 617)}
TOCANTINS = {'min': (2019, 21), 'max': (2005, 271)}
```

Above cell gives us opportunity to see that 'AMAPA' state has less deforested area amoung other states followed by 'TONCANTINS'.

Function to plot a line chart for total deforestation in each year

In [26]:

```
def plot_line_chart_total_deforest():
    fig, axs = plt.subplots(figsize=(10,8))

    axs.plot('year', 'total_deforestation', '-o', data=DEFOREST_DF, color='green', label='Total Deforestation')

    axs.set_title('Total deforestation over the years', fontsize=15)
    axs.set_xlabel('Years', fontsize=14)
    axs.set_ylabel('Deforestation Area ($km^2$)', fontsize=14)
    axs.set_xticks(range(2004, 2020, 1))
    axs.set_yticks(range(0, 29000, 2000))
    axs.set_xlim(left=2003, right=2020)
    axs.set_ylim(bottom=0, top=29000)
    axs.grid(True, axis='both', linestyle='dotted')

    fig.tight_layout()

    return fig
```

Deforestation in each state

Function to plot heatmap of desforesation in each state over the years

In [27]:

```
def plot_heatmap_deforestation():
    states = DEFOREST_DF.columns[1:10]
    states_fullname = [STATES_FULLNAME_DICT[state] for state in states]

    axs = sns.heatmap(DEFOREST_DF.iloc[:,1:10],
                      yticklabels=DEFOREST_DF['year'],
                      xticklabels=states_fullname,
                      cmap='YlGn_r',
                      center=0)

    axs.set_title('Deforestation heatmap for each state over the years', fontsize=15)
    axs.set_xlabel('States', fontsize=12)
    axs.set_ylabel('Year', fontsize=12)

    return
```

Comparing states' contributions

Function to plot pie charts to show the deforestation percentages of the states over the years

In [28]:

```
def plot_grid_pie_chart_state_deforest():
    nrows = 8
    ncols = 2

    fig,axs=plt.subplots(nrows=nrows, ncols=ncols, figsize=(20, 40))
    colors = sns.color_palette('tab10', n_colors=9)
    fig.suptitle('Deforestation percentages of the states from the year 2004 to 2019', x=0.5, y=1.01, fontsize=18)
```

```
    row = 0
    col = 0

    for data_row in DEFOREST_DF.values.tolist():
        if col == ncols:
            col=0
            row+=1

        year_data = data_row[1:10]

        explode = [0]*9

        for i, area in enumerate(year_data):
            if (area/data_row[10]) < 0.02:
                explode[i] = 1 - area/data_row[10]

        axs[row][col].pie(year_data,
                          labels=STATES_FULLNAME_DICT.values(),
                          autopct='%.2f%%',
                          pctdistance=0.7,
                          explode=explode,
                          colors=colors,
                          textprops={'size': 'large'},
                          labeldistance=None,
                          counterclock=False)

        axs[row][col].set_title('Year: ' + str(int(data_row[0])),
                                loc='left', fontsize=15)

        axs[row][col].legend(loc='center left',
                             bbox_to_anchor=(-0.35, .5),
                             fontsize=10, title='BLA States',
                             title_fontsize=12)

        col+=1

    fig.tight_layout()

    return fig
```

Function to plot stacked bar chart to show the contribution of each state to total deforestation in a year

In [29]:

```
def plot_stacked_bar_states_deforest():
    fig, axs = plt.subplots(figsize=(10,8))

    NEXT_STATE_BOTTOM = pd.Series(data=0, index=DEFOREST_DF.index)

    for state in STATES_SHORTNAME_LIST:
       axs.bar(DEFOREST_DF['year'],
               DEFOREST_DF[state],
               bottom=NEXT_STATE_BOTTOM,
               label=STATES_FULLNAME_DICT[state])

       NEXT_STATE_BOTTOM += DEFOREST_DF[state]

    axs.set_title('Individual state contributions to the total deforestation over the years', fontsize=15)
    axs.set_xlabel('Years', fontsize=14)
    axs.set_ylabel('Deforestation Area ($km^2$)', fontsize=14)
    axs.set_xticks(range(2004, 2020, 1))
    axs.set_yticks(range(0, 29000, 2000))
    axs.legend(fontsize=10, title='BLA States', title_fontsize=12)
    axs.grid(True, axis='y', linestyle='dotted')

    fig.tight_layout()

    return fig
```

Analyzing the firespots-dataset

Check the aggregated yearly sum of firespots in each state and in each year

In [30]:

```
agg_firespots_sum_yearly_per_state().iloc[:,:-1]
```

Out[30]:

	year	ACRE	AMAPA	AMAZONAS	MARANHAO	MATO GROSSO	PARA	RONDONIA	RORAIMA	TOCANTINS
0	1999	347	101	1048	4136	28538	20478	7121	220	869
1	2000	430	253	857	4500	17242	18201	5505	362	818
2	2001	829	1300	1297	7979	20795	28590	5062	2415	1408
3	2002	7985	3730	10203	20848	79680	106849	39132	2845	2601
4	2003	10523	2516	10191	12036	50713	53040	30533	3987	861
5	2004	7271	3413	8083	11443	70422	74214	40824	2221	746
6	2005	15993	2020	15644	10820	53489	71477	41641	1461	1175
7	2006	6198	1665	11697	7885	32745	55840	25699	2189	504

	year	ACRE	AMAPA	AMAZONAS	MARANHAO	MATO GROSSO	PARA	RONDONIA	RORAIMA	TOCANTINS
8	2007	8549	1484	11293	12329	52399	68491	27477	3244	1214
9	2008	5699	2153	6701	7822	18602	48449	11549	1950	528
10	2009	3511	2456	9280	6664	9970	41664	5396	2359	382
11	2010	8661	1000	12139	7298	28362	57196	16924	1918	1116
12	2011	3191	1396	5028	4976	9103	26563	6078	1475	376
13	2012	4720	2518	9114	6919	16133	37221	8312	1376	406
14	2013	4980	1529	6512	4418	10830	24046	4613	1395	365
15	2014	4398	1848	9008	5733	15677	35526	7327	2519	518
16	2015	5779	2936	13419	7475	17599	43164	13105	2452	509
17	2016	7684	2595	11173	4928	15836	29724	11462	3870	489
18	2017	6295	1946	11685	5896	18143	49770	11298	1565	841
19	2018	6626	1206	11446	2449	11621	22080	10253	2383	281
20	2019	6802	1272	12665	3989	17479	29700	11206	4775	247

Above cell shows 'PARA' has maximum number of firespots over the years.

Check the aggregated monthly sum of firespots in each state and in each year

In [31]:

```
agg_firespots_sum_monthly_per_state().iloc[:,:10]
```

Out[31]:

	month	ACRE	AMAPA	AMAZONAS	MARANHAO	MATO GROSSO	PARA	RONDONIA	RORAIMA	TOCANTINS
0	JANUARY	63	458	2284	4108	4635	13613	579	8078	87
1	FEBRUARY	33	25	1282	344	4463	1823	299	8457	71
2	MARCH	28	21	771	85	6518	757	328	13497	42
3	APRIL	116	21	333	80	7878	512	352	4607	31
4	MAY	312	22	398	298	17417	1439	826	542	77
5	JUNE	640	44	1391	1030	44853	7053	3003	128	246
6	JULY	4712	72	13010	2961	56567	45978	14644	108	668
7	AUGUST	36083	402	63558	7049	139145	241683	93012	316	2686
8	SEPTEMBER	70313	2741	59857	14570	206432	208204	162965	1019	7527
9	OCTOBER	12871	11143	28676	35117	82499	148781	53842	2371	3620
10	NOVEMBER	1217	17774	12770	57652	17490	183887	8633	3625	912
11	DECEMBER	83	6614	4153	37249	7481	88553	2034	4233	287

Above cell shows increase in number of fire outbreaks from July to December

Check the yearly average firespots in each year

In [32]:

```
agg_firespots_sum_monthly_per_state()[['month', 'yearly_avg_firespots']]
```

Out[32]:

	month	yearly_avg_firespots
0	JANUARY	1695.25
1	FEBRUARY	839.85
2	MARCH	1102.35
3	APRIL	696.50
4	MAY	1066.55
5	JUNE	2919.40
6	JULY	6936.00
7	AUGUST	29196.70
8	SEPTEMBER	36681.40
9	OCTOBER	18946.00
10	NOVEMBER	15198.00
11	DECEMBER	7534.35

Above cell also shows the yearly average of fire outbreaks are maximum in the months starting from July till December

Total firespots in each year

Function to plot a bar chart to show the total firespots aggregated over the months for each year

In [33]:

```
def plot_line_chart_total_firespots():
    fig, axs = plt.subplots(figsize=(12,8))

    YEARLY_FIRESPOTS_DF = agg_firespots_sum_yearly_per_state()

    axs.bar(YEARLY_FIRESPOTS_DF['year'],
            YEARLY_FIRESPOTS_DF['total_firespots'],
            width=0.6,
            color='red',
            alpha=0.5,
            label='Total Firespots')

    axs.set_title('Total firespots over the years', fontsize=15)
    axs.set_xlabel('Years', fontsize=14)
    axs.set_ylabel('Firespots', fontsize=14)
    axs.set_xticks(range(1999, 2020, 1))
    axs.set_yticks(range(0, 290000, 10000))
    axs.set_xlim(left=1998, right=2020)
    axs.set_ylim(bottom=0, top=290000)
    axs.grid(True, axis='y', linestyle='dotted')

    fig.tight_layout()

    return fig
```

Yearly firespots for each state

Function to plot line charts of aggregated sum of firespots for each state over the years.

In [34]:

```
def plot_line_chart_firespots_states():
    YEARLY_FIRESPOTS_DF = agg_firespots_sum_yearly_per_state()

    xmin = int(YEARLY_FIRESPOTS_DF['year'].min())
    xmax = int(YEARLY_FIRESPOTS_DF['year'].max())
    ymin = int(YEARLY_FIRESPOTS_DF.iloc[:,1:10].min().min())
    ymax = int(YEARLY_FIRESPOTS_DF.iloc[:,1:10].max().max())

    axs = YEARLY_FIRESPOTS_DF.plot(kind='line',
                                   x='year',
                                   y=YEARLY_FIRESPOTS_DF.columns[1:10],
                                   colormap='tab10',
                                   figsize=(15, 10),
                                   lw=1.5)

    axs.set_title('No. of firespots recorded for each state over the years', fontsize=15)
    axs.set_xlabel('Years', fontsize=14)
    axs.set_ylabel('Firespots', fontsize=14)
    axs.set_xticks(range(xmin, xmax+1, 1))
    axs.set_yticks(range(0, ymax+5000, 5000))
    axs.set_xlim(left=xmin-1, right=xmax+1)
    axs.set_ylim(bottom=0, top=ymax+5000)
    axs.grid(True, axis='y', linestyle='dotted')
    axs.legend(fontsize=12, title='BLA States', title_fontsize=12)

    return axs
```

Monthly firespots for each state

Function to plot heatmap of fire outbreaks in each state aggreagated over the years for each month

In [35]:

```
def plot_heatmap_fires():
    MONTHLY_FIRESPOTS_DF = agg_firespots_sum_monthly_per_state()

    axs = sns.heatmap(MONTHLY_FIRESPOTS_DF.iloc[:,1:10],
                      yticklabels=MONTHLY_FIRESPOTS_DF['month'],
                      xticklabels=MONTHLY_FIRESPOTS_DF.columns[1:10],
                      cmap = 'coolwarm',
                      center=0)

    axs.set_title('Firespots heatmap for each state over the months', fontsize=15)
    axs.set_xlabel('States', fontsize=12)
    axs.set_ylabel('Months', fontsize=12)

    return axs
```

Monthly average firespots

Function to plot line chart to show yearly average of the firespots per month

In [36]:

```
def plot_line_chart_months_yearly_avg_fires():
    MONTHLY_FIRESPOTS_DF = agg_firespots_sum_monthly_per_state()

    fig, axs = plt.subplots(figsize=(10,8))

    axs.plot(MONTHLY_FIRESPOTS_DF['month'],
             MONTHLY_FIRESPOTS_DF['yearly_avg_firespots'],
             label='Yearly Avg Firespots',
             marker='o')
```

```
    axs.set_title('Yearly average of fire outbreaks in each month', fontsize=15)
    plt.xticks(rotation = 90)
    axs.set_xlabel('Months', fontsize=14)
    axs.set_ylabel('Firespots', fontsize=14)
    axs.set_yticks(range(0, 40000, 2000))
    axs.grid(True, axis='y', linestyle='dotted')

    fig.tight_layout()

    return fig
```

Correlating the datasets

Total deforestation versus total firespots

Function to plot shared graph for total deforestation and total fire oubreaks over the years

In [37]:

```
def plot_total_deforest_v_fires():
    fig, ax = plt.subplots(figsize=(12,6))

    ax.bar(DEFOREST_DF['year'],
           DEFOREST_DF['total_deforestation'],
           label='Total Deforestation',
           color='green',
           alpha=0.5)

    ax2 = ax.twinx() # create a second Axes sharing the x axis of ax

    YEARLY_FIRESPOTS_DF = agg_firespots_sum_yearly_per_state()
    YEARLY_FIRESPOTS_DF = YEARLY_FIRESPOTS_DF[YEARLY_FIRESPOTS_DF['year']>=2004]

    ax2.plot(YEARLY_FIRESPOTS_DF['year'],
             YEARLY_FIRESPOTS_DF['total_firespots'],
             label='Total Firespots',
             color='red',
             alpha=0.8,
             lw=1.2)

    ax.set_xticks(range(2004, 2019, 1))
    ax.set_yticks(range(0, 32000, 2000))
    ax.set_xlim(left=2003, right=2020)
    ax.set_ylim(bottom=0, top=32000)

    ax2.set_xticks(range(2004, 2020, 1))
    ax2.set_yticks(range(0, 320000, 20000))
    ax2.set_xlim(left=2003, right=2020)
    ax2.set_ylim(bottom=0, top=320000)

    # add titles and labels
    ax.set_title('Total deforestation vs total firespots', fontsize=15)

    ax.set_ylabel('Deforestation Area ($km^2$)', fontsize=12)
    ax2.set_ylabel('Firespots', fontsize=12)
    ax.set_xlabel('Years', fontsize=12)

    ax.grid(True, axis='y', linestyle='dotted')

    ax.legend(fontsize=12, loc='upper left')
    ax2.legend(fontsize=12, loc='upper right')

    fig.tight_layout()

    return fig
```

Project Outcome (10 + 10 marks)

Overview of Results

Objective 1

To analyze the trend of the deforestation in the states of the Amazon rainforest in Brazil.

Explanation of Results

The total deforestation plotted in Figure1 shows that maximum deforestation of 27772 km^2 was seen in the year 2004, and minimum deforestation of 4571 km^2 was seen in the year 2012. From the year 2004 to 2012, deforestation has plummeted drastically whereas, from 2012 onwards, the total deforestation is slightly increasing.

In the given period, as evident in Figure2, the worst deforestation is observed in the Mato Grosso state in the year 2004. However, the average deforested area of Para has been quite extensive over the years.

Figure3 indicates that Mato Grosso, Para, and Rondonia states are the worst affected by deforestation. On the one hand, the contribution of Mato Grasso to the total deforestation significantly dropped over the years from 42.54% to 17.26%. On the other hand, the contribution of Para and Amazonas increased from 31.94% and 4.4% to 39.56% and 14.56%, respectively. In addition, the contribution of Rondonia didn't drop significantly over the years.

It can be concluded from Figure4 that Acre, Amazonas, Para and Rondonia are the four states which closely follow the total deforestation trend shown in Figure1. Additionally, as deforestation has been increasing since 2012, deforestation in these four states is more likely to increase in the coming years. Apart from this, in the year 2008, there was a sudden rise in deforestation due to the drastic increase of deforestation in Maranhao state.

Visualisation

Figure1 shows the trend of total deforestation in km^2 of the BLA states over the years.

In [38]:

```
Figure1 = plot_line_chart_total_deforest()
Figure1.show()
```

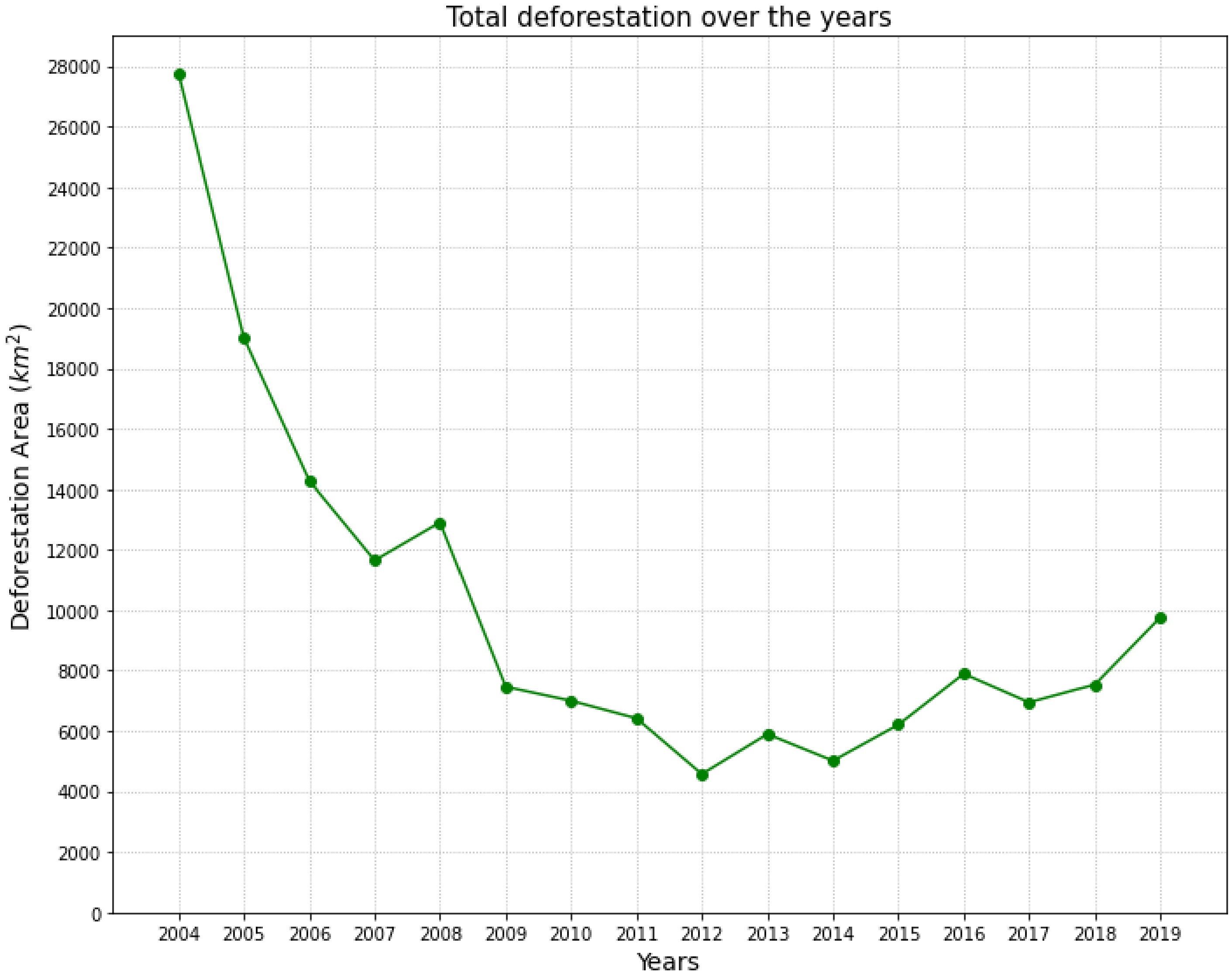


Figure2 shows the heatmap of deforestation in km^2 of the BLA states over the years.

In [39]:

```
Figure2=plot_heatmap_deforestation()
```

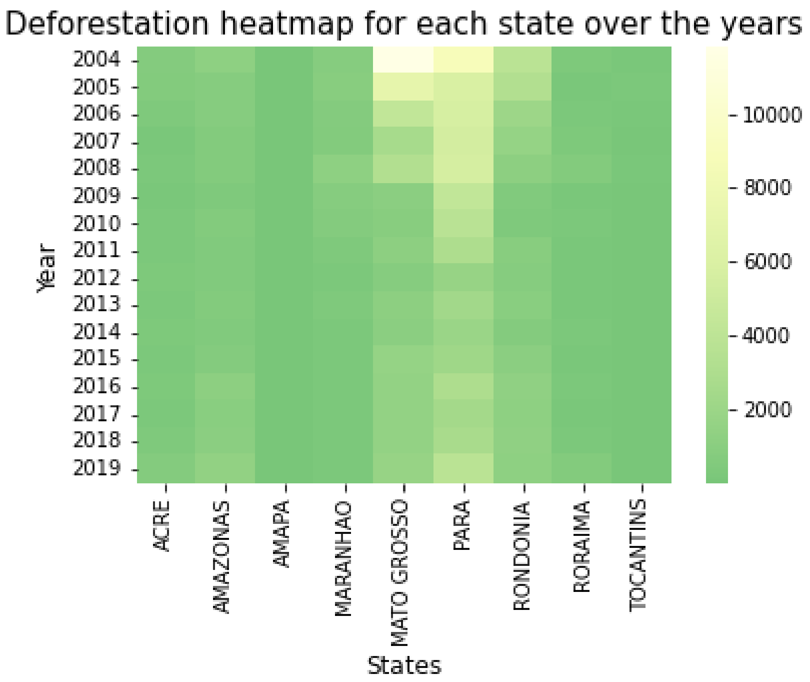


Figure3 shows the percentage contributions of the BLA states to the total deforestation in each year.

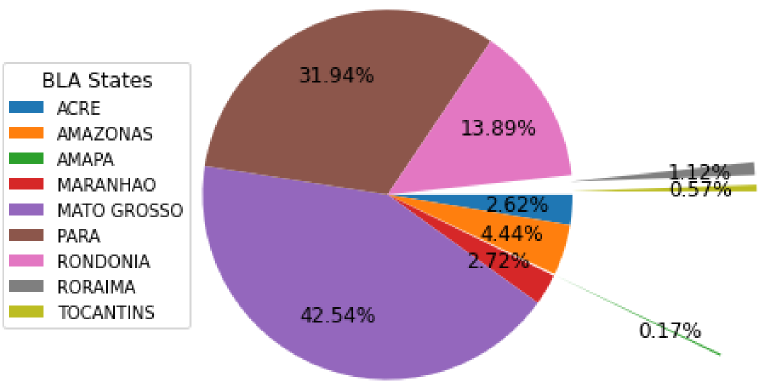
In [40]:

```
Figure3 = plot_grid_pie_chart_state_deforest()
Figure3.show()
```

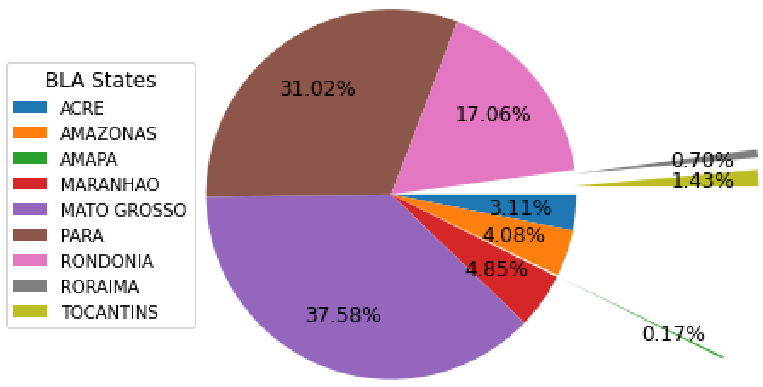
Deforestation percentages of the states from the year 2004 to 2019

Year: 2004

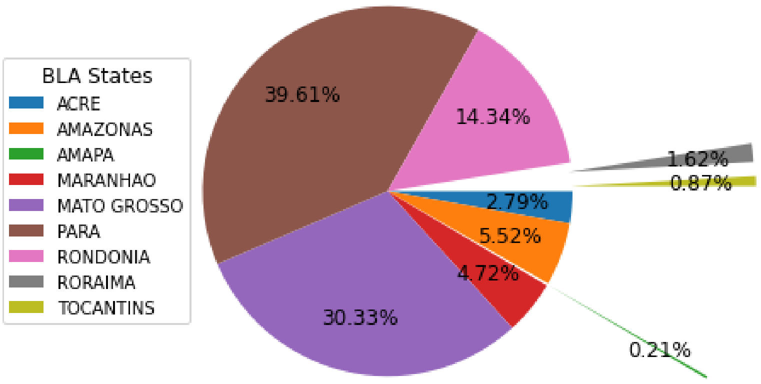
Year: 2005



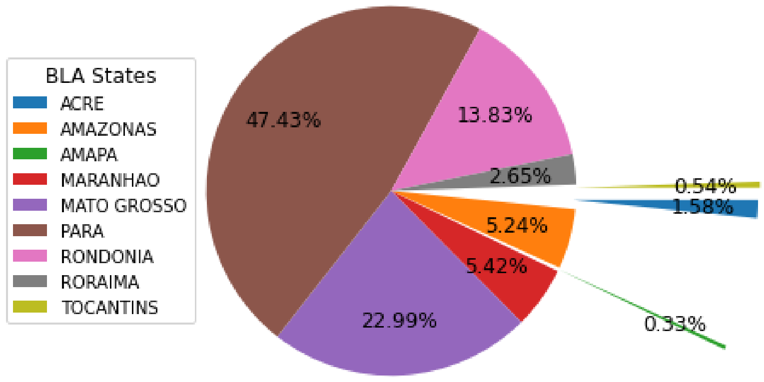
Year: 2006



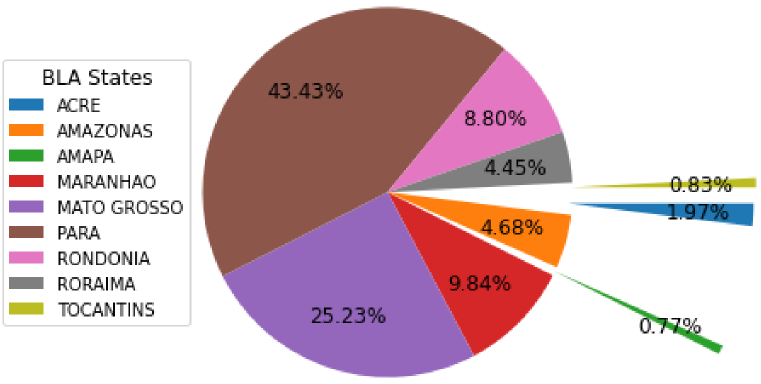
Year: 2007



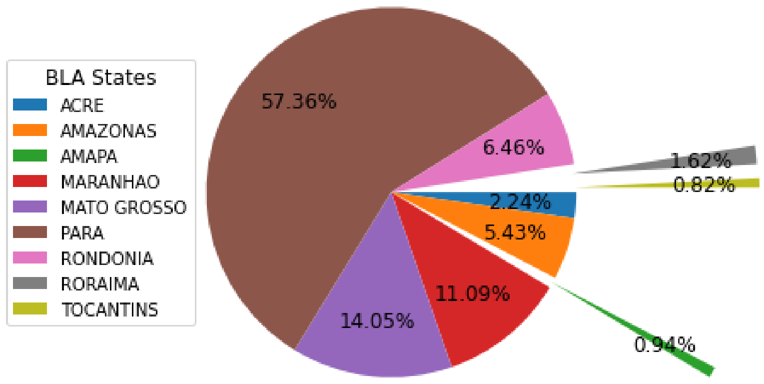
Year: 2008



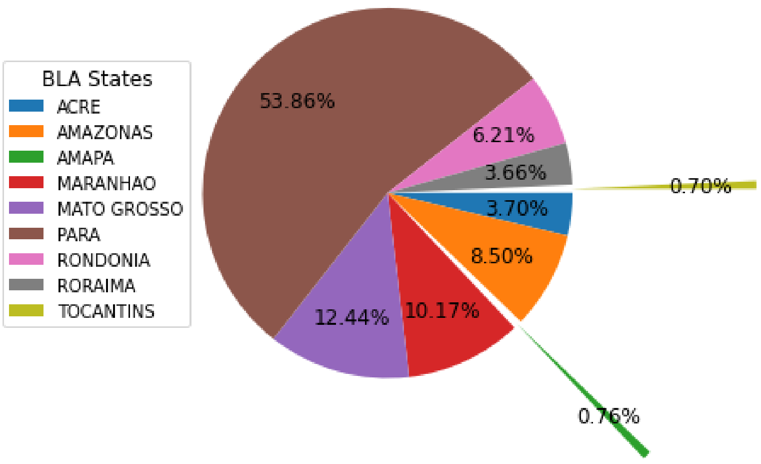
Year: 2009



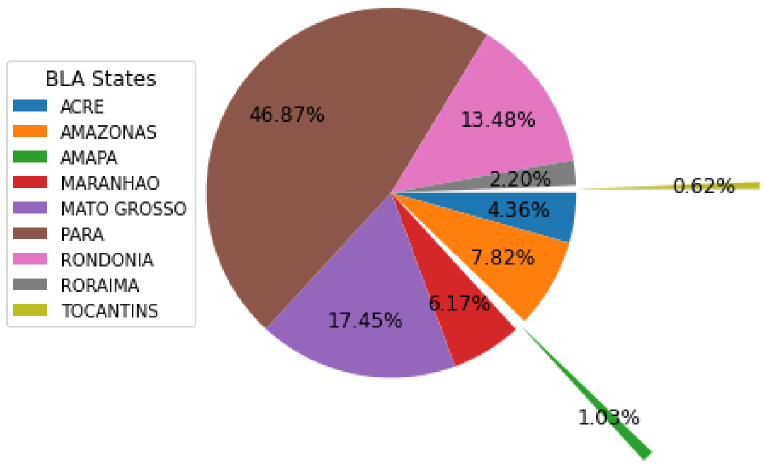
Year: 2010



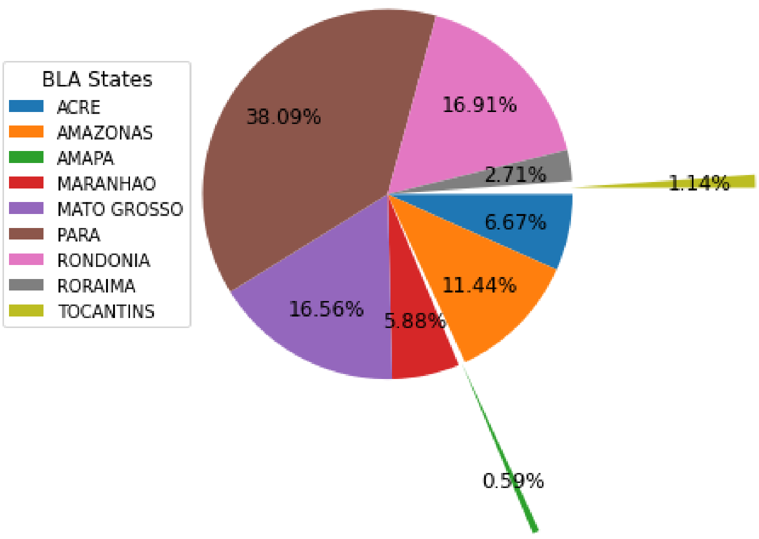
Year: 2011



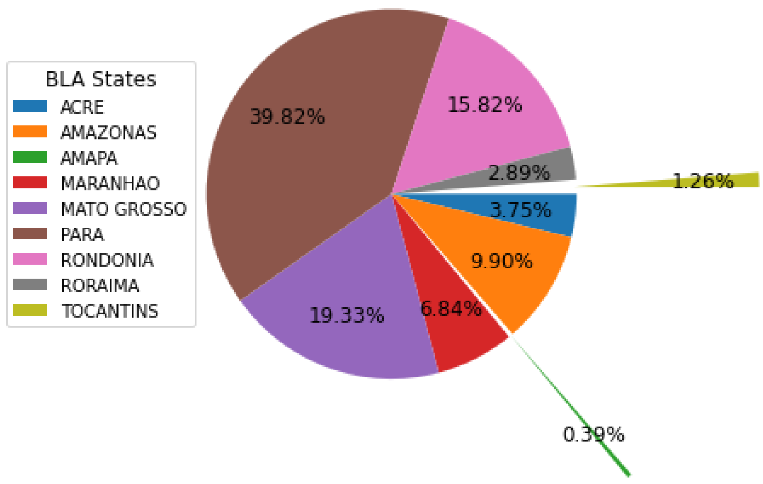
Year: 2012



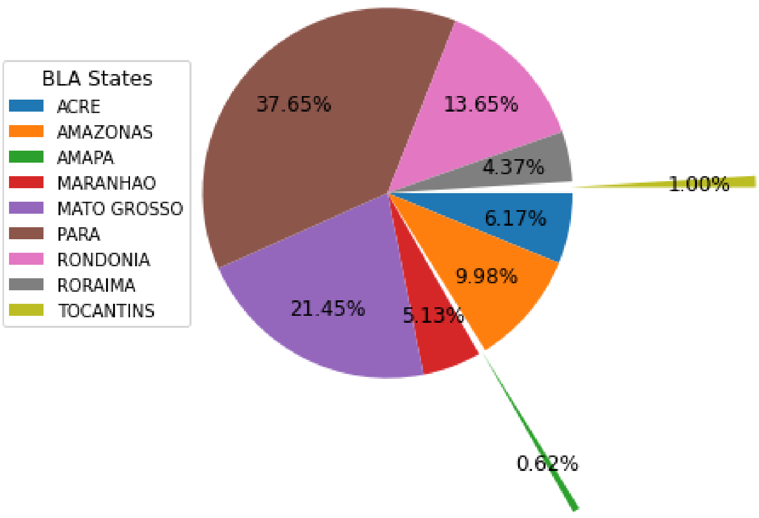
Year: 2013

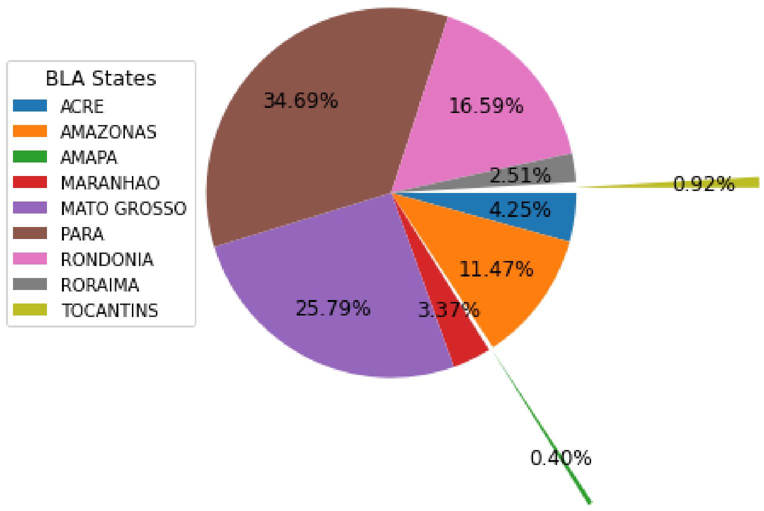


Year: 2014

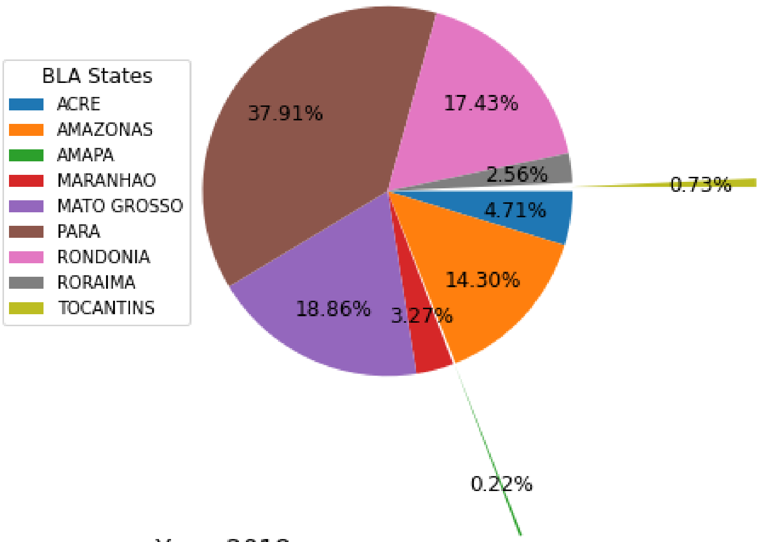


Year: 2015

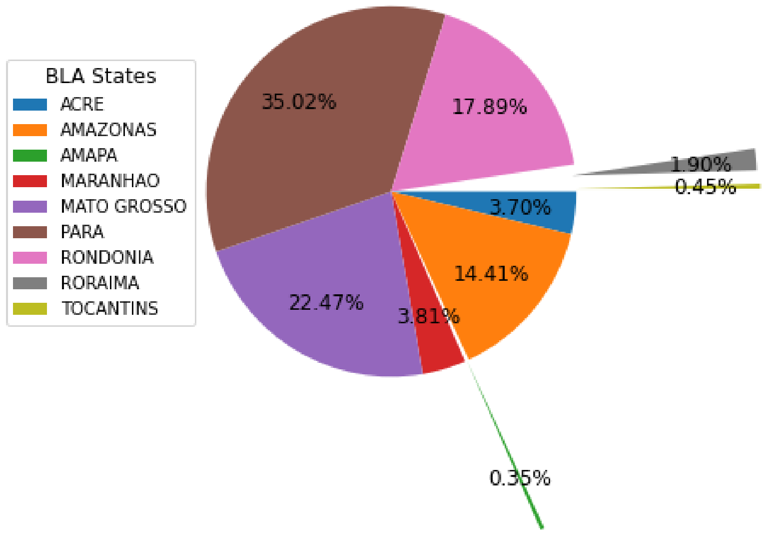




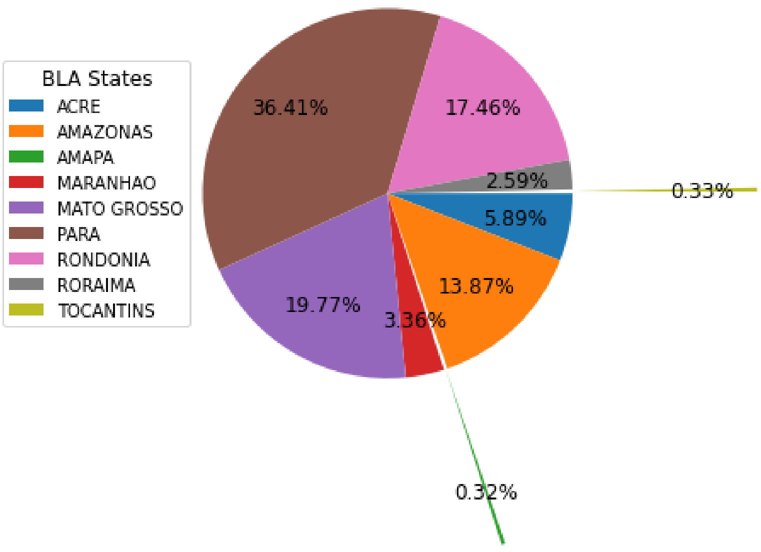
Year: 2016



Year: 2017



Year: 2018



Year: 2019

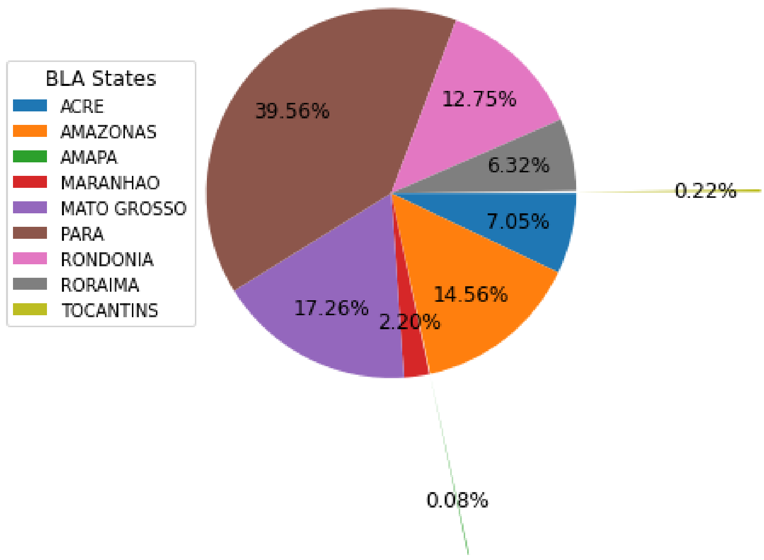
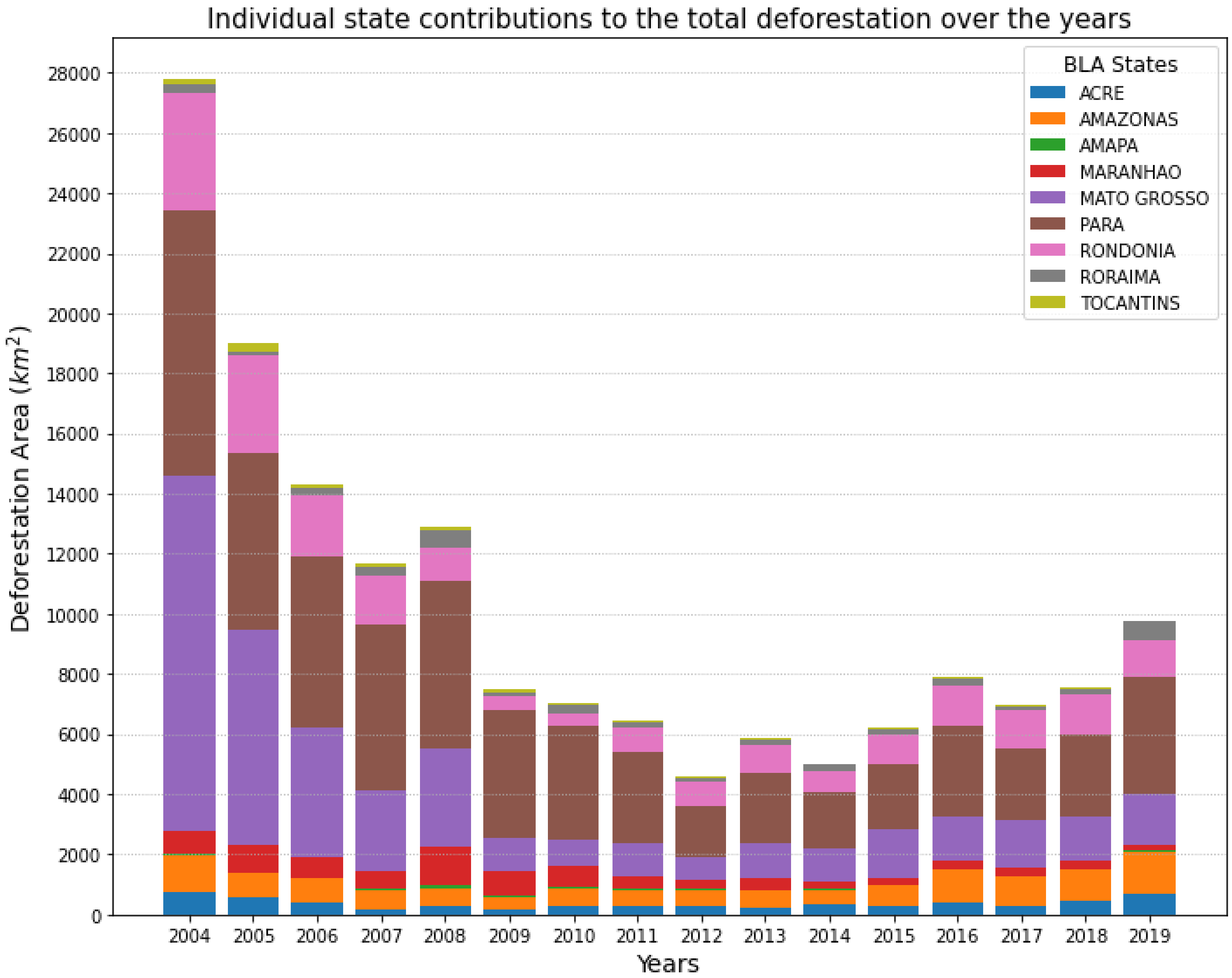


Figure4 shows individual state contributions to the total deforestation in km^2 over the years.

In [41]:
```
Figure4=plot_stacked_bar_states_deforest()
Figure4.show()
```



Objective 2

To analyze the occurance of fire outbreaks in the states of the Amazon rainforest in Brazil.

Explanation of Results

In Figure5, the maximum fire outbreaks(firespots) of 273873 are seen in the year 2002 and the minimum fire outbreaks of 48168 are seen in the year 2000. Nearly a four-fold increase in the number of fire outbreaks is observed from the year 2001 to 2002. In contrast, the number of fire outbreaks dropped by almost 50% from the year 2010 to 2011. In recent years, after 2011 there is a slight increase in the number of fire outbreaks.

From 2001 to 2008, as indicated in Figure6, Para, Mato Grosso and Rondonia outstripped all other states in the number of fire outbreaks. However, the number of fire outbreaks in Mato Grosso and Rondonia reduced drastically after 2008, whereas in Para, the number of fire outbreaks continued to be high. Additionally, it can be observed that all the nine BLA states showed similar variations in the number of fire outbreaks from 1999 to 2019.

An interesting fact is observed in the heatmap shown in Figure7 and the plot shown in Figure8. That is in the months of August, September and October, the largest number of fire outbreaks are recorded. This might be due to the high temperatures during the dry season in Brazil from August to October. In addition, more than 200000 fire outbreaks happened in Para only in the month of August in a span of 20 years. However, the maximum number of fire outbreaks are seen in the month of September on an average.

Visualisation

Figure5 shows total firespots observed over 20 years from 1999 to 2019.

In [42]:

```
Figure5=plot_line_chart_total_firespots()
Figure5.show()
```

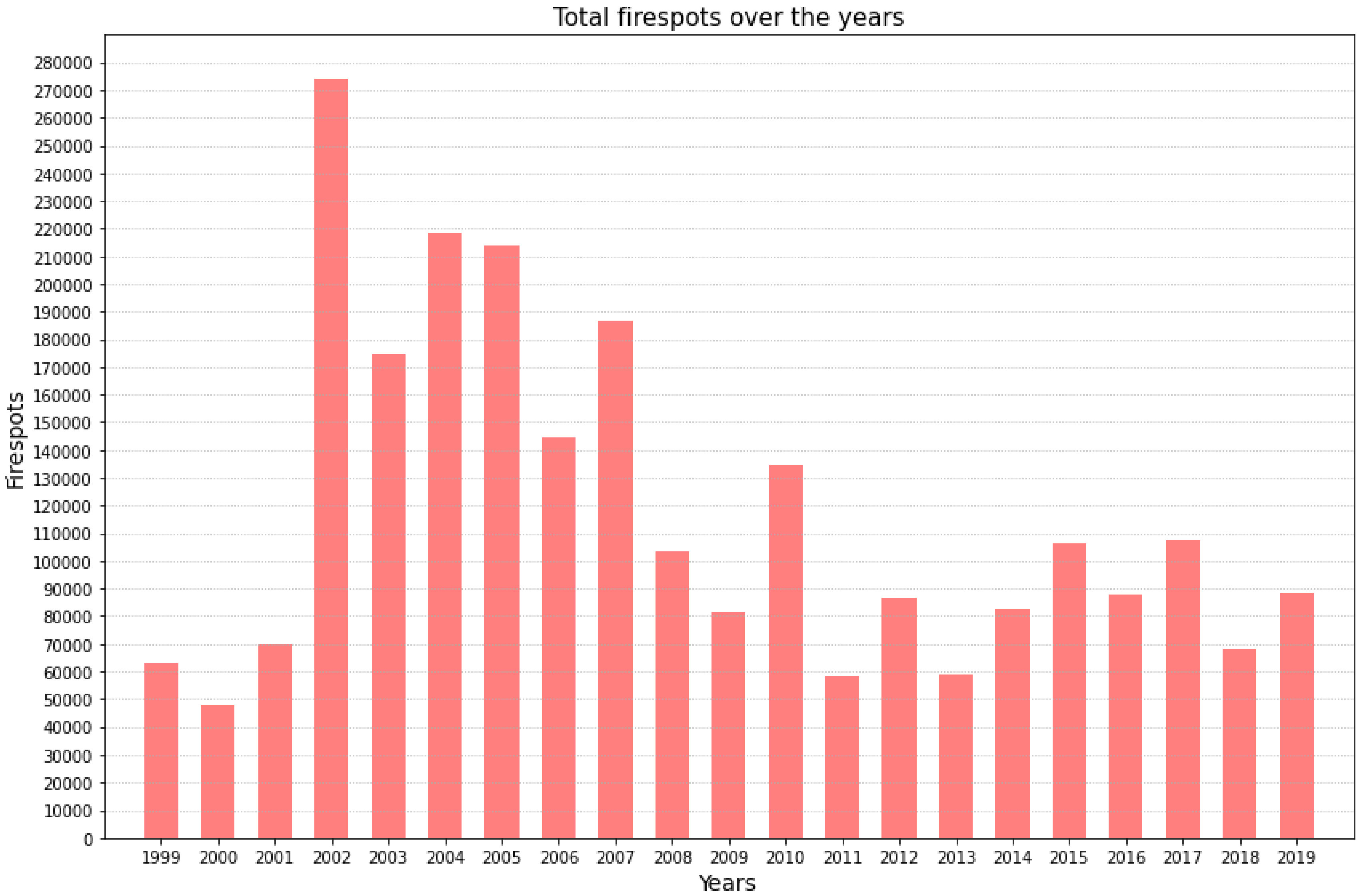


Figure6 shows the number of firespots observed for each state over 20 years from 1999 to 2019.

In [43]:

```
Figure6=plot_line_chart_firespots_states()
```

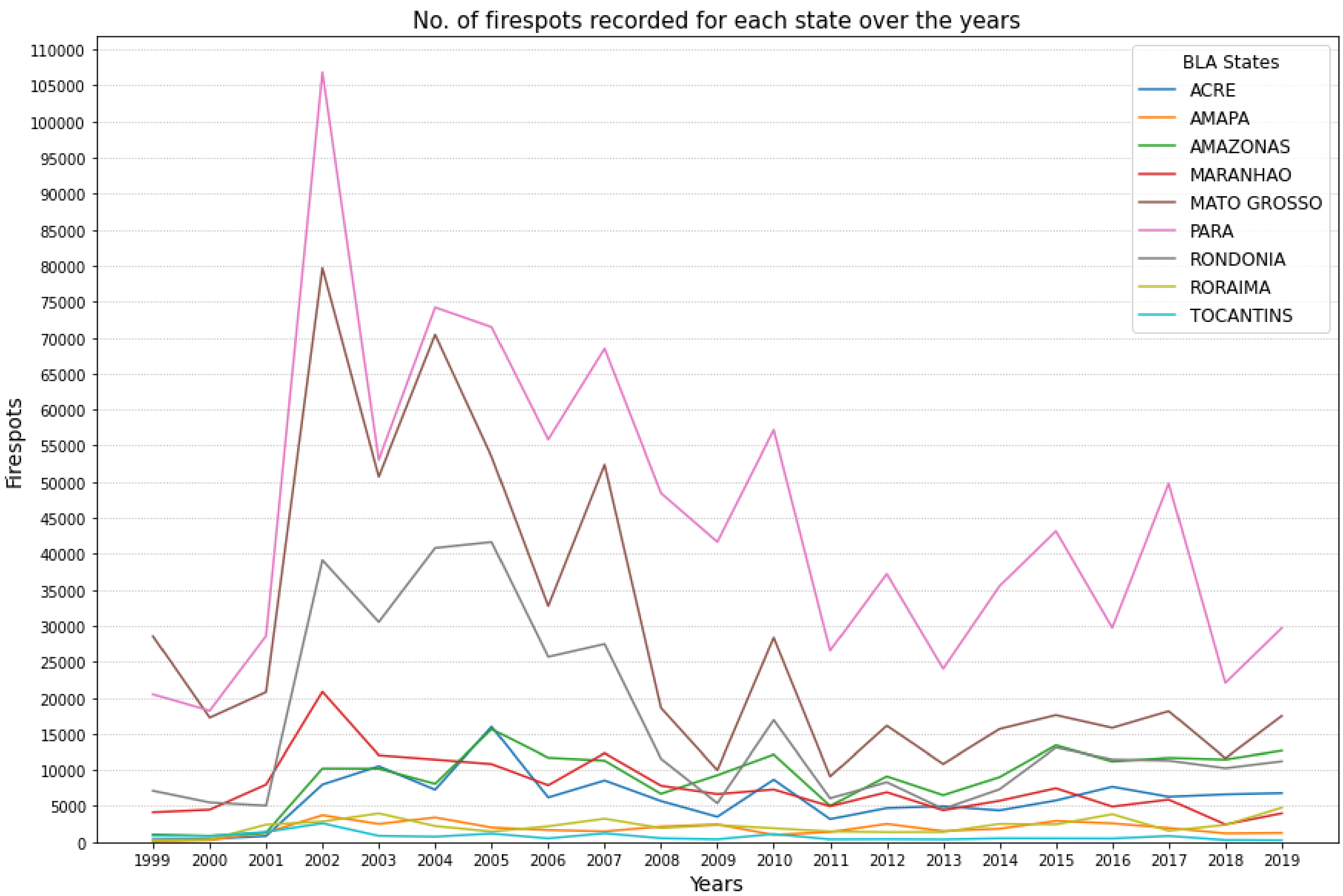


Figure7 shows the heatmap of the total aggregated firespots in each state and in each month.

In [44]:

```
Figure7=plot_heatmap_fires()
```

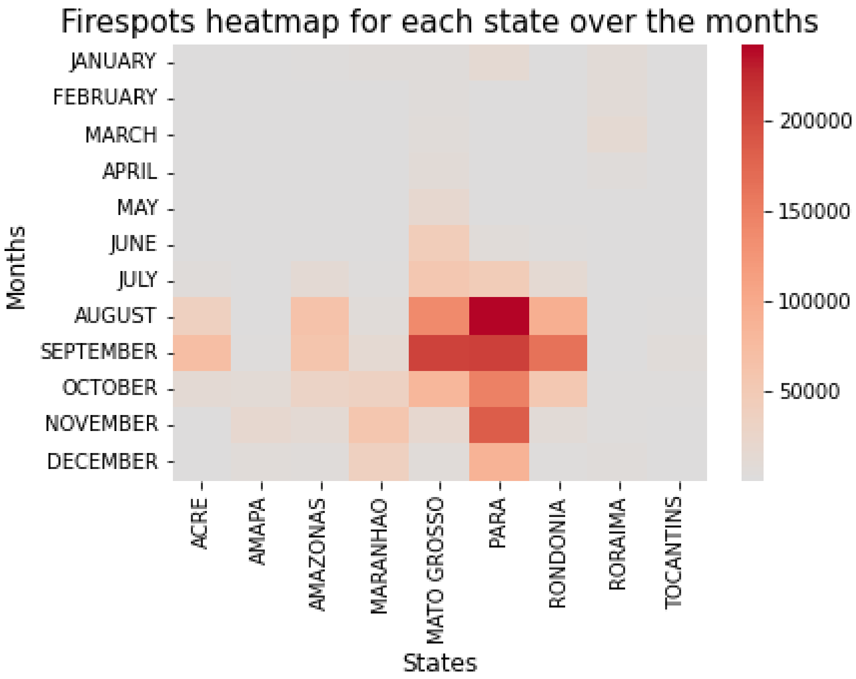
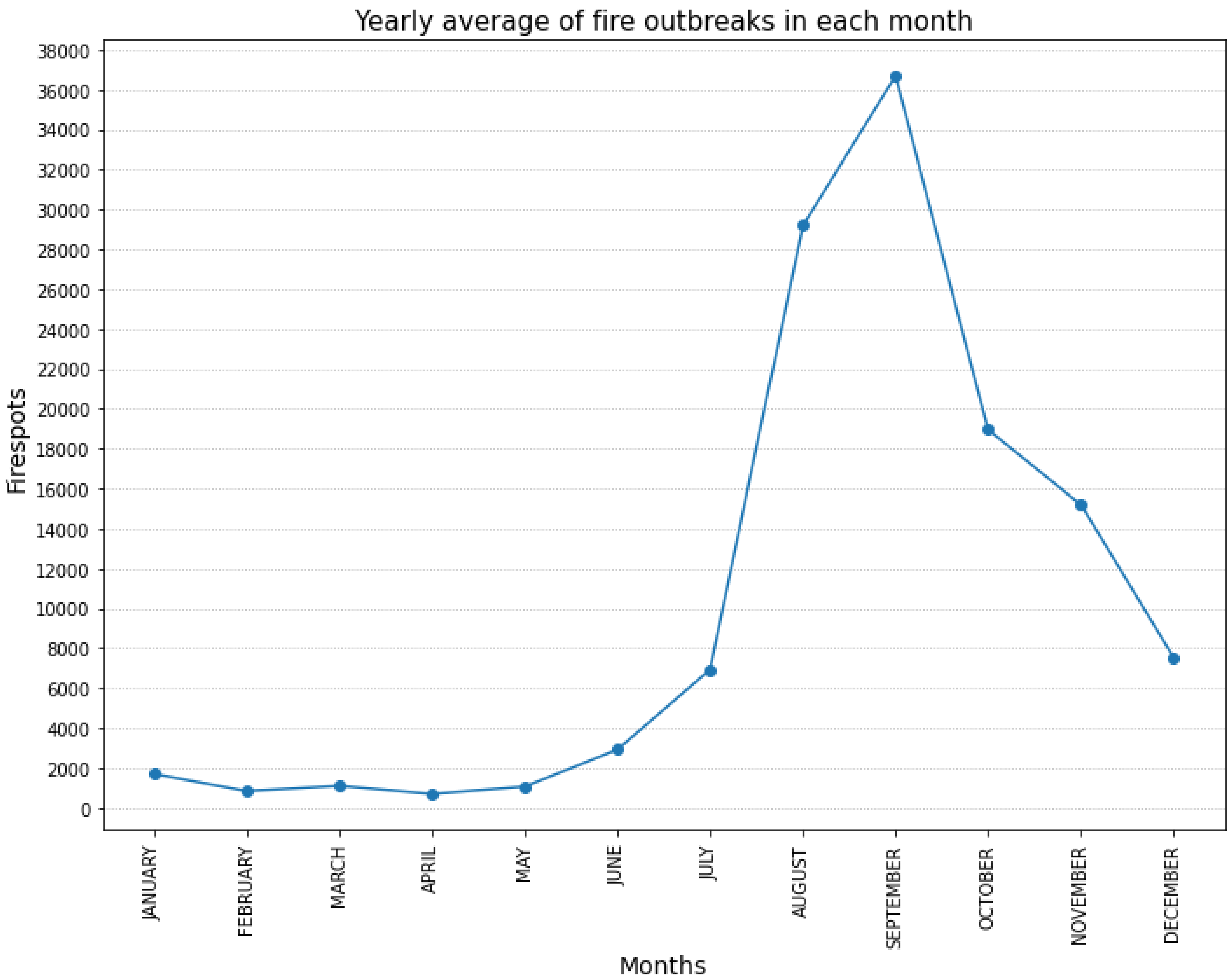


Figure8 shows the trend of yearly average of firespots over the months.

In [45]:
```
Figure8=plot_line_chart_months_yearly_avg_fires()
```



Objective 3

To correlate the deforestation data and the number of fire outbreaks in the states of the Amazon rainforest in Brazil.

Explanation of Results

It can be concluded from Figure9 that even though there are some fluctuations in the fire outbreaks over the years, roughly the total deforestation and the total firespots follow the same trend from 2004 to 2019.

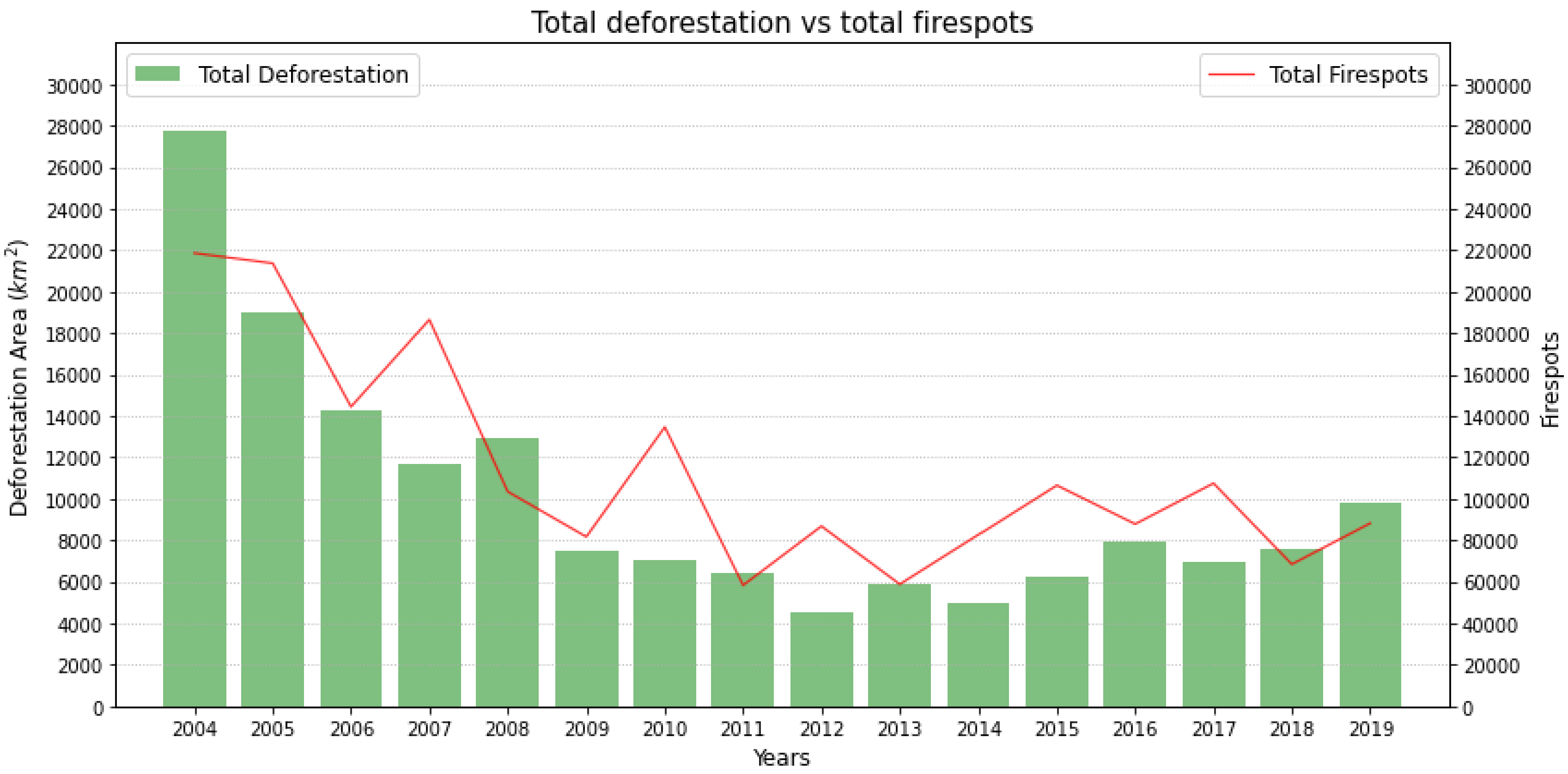
Moreover, from Figure4 and Figure6, it is clear that Para, Mato Grosso and Rondonia had the highest number of fire outbreaks and the largest deforestation areas which strongly supports the above conclusion.

Last but not the least, there are some occasions (e.g., 2007, 2010, 2012, etc.) when deforestation has reduced yet the number of fire outbreaks has increased. Such scenarios are difficult to explain due to a lack of additional information.

Visualisation

Figure9 shows the correlation between the trends of total deforestation and total fire outbreaks observed during 2004 to 2019

In [46]:
```
Figure9 = plot_total_deforest_v_fires()
Figure9.show()
```



Conclusion (5 marks)

Acheivements

The deforestation and the firespots datasets are analyzed to understand the trends of deforestation and fire outbreaks in the states of the Amazon rainforest in Brazil. Among the nine BLA states, **Para** has suffered from the worst deforestation and has seen the highest number of fire outbreaks from 2004 to 2019. In addition, the dry season (June-December) of Brazil witnessed most of the fire outbreaks over the years in the Amazon rainforest region. When the deforestation data and firespots data are correlated, both are found to follow a roughly similar trend.

Limitations

One of the limitations of this project is that there is no record of the deforestation data from the year 1999 to 2003. Because of this firespots data for the same period could not be used to analyze the correlation between the deforestation and the firespots datasets.

Another limitation is the unavailability of multiple feature vectors - apart from the firespots - contributing to deforestation. As result, the fluctuations in deforestation and the firespots could not be explained accurately.

Future Work

Machine learning algorithms and neural network models can be used to efficiently analyze the used dataset and accurately predict the future deforestation and fire outbreaks in the BLA states.

```
In [ ]:
```