Pandas

- Pandas is a Python library used for working with data sets.
- It has functions for analyzing, cleaning, exploring, and manipulating data.

Uses

- Pandas allows us to analyze data and make conclusions based on statistical theories.
- Pandas can clean messy data sets, and make them readable and relevant.
- Relevant data is very important in data science.

Import Pandas module

- `Import` = “Bring this functionality or library to my python script”
- `Pandas` = The library you want to import, in this case, it’s pandas
- `As` = The python nomenclature for creating as alias. This is a fancy way of taking a long word and referencing it as a short word
- `pd` = The standard short name for referencing pandas

In [4]:

```python
import pandas as pd
#import matplotlib.pyplot as plt
```

Load a CSV file into a Pandas DataFrame

- Pandas read_csv() function imports a CSV file to DataFrame format.
- A Pandas DataFrame is a 2 dimensional data structure.

In [5]:

```
dataset = pd.read_csv('amazon.csv',encoding="latin-1")
dataset
```

Out[5]:

	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
...	...	...	...	...	...
6449	2012	Tocantins	Dezembro	128.0	2012-01-01
6450	2013	Tocantins	Dezembro	85.0	2013-01-01
6451	2014	Tocantins	Dezembro	223.0	2014-01-01
6452	2015	Tocantins	Dezembro	373.0	2015-01-01
6453	2016	Tocantins	Dezembro	119.0	2016-01-01

6454 rows × 5 columns

In [25]:

```
type(dataset)
```

Out[25]:

```
pandas.core.frame.DataFrame
```

In [19]:

```
dataset.shape
```

Out[19]:

```
(6454, 5)
```

In [20]:

```
dataset.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
```

In [27]:

```
dataset.head(10)
```

Out[27]:

	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
5	2003	Acre	Janeiro	10.0	2003-01-01
6	2004	Acre	Janeiro	0.0	2004-01-01
7	2005	Acre	Janeiro	12.0	2005-01-01
8	2006	Acre	Janeiro	4.0	2006-01-01
9	2007	Acre	Janeiro	0.0	2007-01-01

In [29]:

```
dataset.tail(8)
```

Out[29]:

	year	state	month	number	date
6446	2009	Tocantins	Dezembro	46.0	2009-01-01
6447	2010	Tocantins	Dezembro	72.0	2010-01-01
6448	2011	Tocantins	Dezembro	105.0	2011-01-01
6449	2012	Tocantins	Dezembro	128.0	2012-01-01
6450	2013	Tocantins	Dezembro	85.0	2013-01-01
6451	2014	Tocantins	Dezembro	223.0	2014-01-01
6452	2015	Tocantins	Dezembro	373.0	2015-01-01
6453	2016	Tocantins	Dezembro	119.0	2016-01-01

In [22]:

```
dataset.columns
```

Out[22]:

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

In [10]:

```
dataset.describe()
```

Out[10]:

	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

In [11]:
```
dataset["year"]
```

Out[11]:
```
0       1998
1       1999
2       2000
3       2001
4       2002
        ... 
6449    2012
6450    2013
6451    2014
6452    2015
6453    2016
Name: year, Length: 6454, dtype: int64
```

In [32]:
```
dataset[["year","state"]]
```

Out[32]:

	year	state
0	1998	Acre
1	1999	Acre
2	2000	Acre
3	2001	Acre
4	2002	Acre
...	...	...
6449	2012	Tocantins
6450	2013	Tocantins
6451	2014	Tocantins
6452	2015	Tocantins
6453	2016	Tocantins

6454 rows × 2 columns

In [33]:
```
dataset.head()
```

Out[33]:

	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

In [16]:
```
dataset.iloc[0]
```

Out[16]:
```
year            1998
state           Acre
month        Janeiro
number           0.0
date      1998-01-01
Name: 0, dtype: object
```

In [18]:
```
dataset.iloc[3:5]
```

Out[18]:

	year	state	month	number	date
3	2001	Acre	Janeiro	0.0	2001-01-01
4	2002	Acre	Janeiro	0.0	2002-01-01

In [38]:
```
dataset.iloc[1:11:2]
```

Out[38]:

	year	state	month	number	date
1	1999	Acre	Janeiro	0.0	1999-01-01
3	2001	Acre	Janeiro	0.0	2001-01-01
5	2003	Acre	Janeiro	10.0	2003-01-01
7	2005	Acre	Janeiro	12.0	2005-01-01
9	2007	Acre	Janeiro	0.0	2007-01-01

In [29]:
```
dataset.iloc[[2,6,19]]
```

Out[29]:

	year	state	month	number	date
2	2000	Acre	Janeiro	0.0	2000-01-01
6	2004	Acre	Janeiro	0.0	2004-01-01
19	2017	Acre	Janeiro	0.0	2017-01-01

In [30]:
```
# will fetch all rows but only col 0, 1
dataset.iloc[:,[0,1]]
```

Out[30]:

	year	state
0	1998	Acre
1	1999	Acre
2	2000	Acre
3	2001	Acre
4	2002	Acre
...	...	...
6449	2012	Tocantins
6450	2013	Tocantins
6451	2014	Tocantins
6452	2015	Tocantins
6453	2016	Tocantins

6454 rows × 2 columns

In [31]:

```
# will fetch all rows but only col 0, 1
dataset.iloc[:11,0:3]
```

Out[31]:

	year	state	month
0	1998	Acre	Janeiro
1	1999	Acre	Janeiro
2	2000	Acre	Janeiro
3	2001	Acre	Janeiro
4	2002	Acre	Janeiro
5	2003	Acre	Janeiro
6	2004	Acre	Janeiro
7	2005	Acre	Janeiro
8	2006	Acre	Janeiro
9	2007	Acre	Janeiro
10	2008	Acre	Janeiro

In [34]:

```
dataset.iloc[[4,6,8],0:3]
```

Out[34]:

	year	state	month
4	2002	Acre	Janeiro
6	2004	Acre	Janeiro
8	2006	Acre	Janeiro

In [36]:

```
dataset.iloc[[44,66,88],[0,2,3]]
```

Out[36]:

	year	month	number
44	2002	Março	0.0
66	2004	Abril	2.0
88	2006	Maio	8.0

loc

- It selects rows and columns with specific labels

In [37]:

```
dataset.loc[0]
```

Out[37]:

```
year                1998
state               Acre
month            Janeiro
number               0.0
date          1998-01-01
Name: 0, dtype: object
```

In [40]:
```
dataset.loc[0:7]
```

Out[40]:

	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
5	2003	Acre	Janeiro	10.0	2003-01-01
6	2004	Acre	Janeiro	0.0	2004-01-01
7	2005	Acre	Janeiro	12.0	2005-01-01

In [41]:
```
dataset.loc[0:7,"year":"month"]
```

Out[41]:

	year	state	month
0	1998	Acre	Janeiro
1	1999	Acre	Janeiro
2	2000	Acre	Janeiro
3	2001	Acre	Janeiro
4	2002	Acre	Janeiro
5	2003	Acre	Janeiro
6	2004	Acre	Janeiro
7	2005	Acre	Janeiro

In [47]:
```
dataset.loc[10:17,["year","month"]]
```

Out[47]:

	year	month
10	2008	Janeiro
11	2009	Janeiro
12	2010	Janeiro
13	2011	Janeiro
14	2012	Janeiro
15	2013	Janeiro
16	2014	Janeiro
17	2015	Janeiro

In [48]:
```
dataset.loc[[10,15,17],["year","month"]]
```

Out[48]:

	year	month
10	2008	Janeiro
15	2013	Janeiro
17	2015	Janeiro

In [50]:
```
dataset["state"].value_counts()
```

Out[50]:
```
Rio                 717
Paraiba             478
Mato Grosso         478
Alagoas             240
Acre                239
Sergipe             239
Sao Paulo           239
Santa Catarina      239
Roraima             239
Rondonia            239
Piau                239
Pernambuco          239
Minas Gerais        239
Pará                239
Maranhao            239
Goias               239
Espirito Santo      239
Distrito Federal    239
Ceara               239
Bahia               239
Amazonas            239
Amapa               239
Tocantins           239
Name: state, dtype: int64
```

Boolean indexing

- In boolean indexing, we use a boolean vector to filter the data i.e True and Flase.

In [42]:
```
mask=dataset["state"]=="Rio"
mask
```

Out[42]:
```
0       False
1       False
2       False
3       False
4       False
        ...
6449    False
6450    False
6451    False
6452    False
6453    False
Name: state, Length: 6454, dtype: bool
```

In [44]:
```
dataset[dataset["state"]=="Rio"]
```

Out[44]:

	year	state	month	number	date
4303	1998	Rio	Janeiro	0.0	1998-01-01
4304	1999	Rio	Janeiro	0.0	1999-01-01
4305	2000	Rio	Janeiro	0.0	2000-01-01
4306	2001	Rio	Janeiro	0.0	2001-01-01
4307	2002	Rio	Janeiro	0.0	2002-01-01
...	...	...	...	...	...
5015	2012	Rio	Dezembro	38.0	2012-01-01
5016	2013	Rio	Dezembro	62.0	2013-01-01
5017	2014	Rio	Dezembro	31.0	2014-01-01
5018	2015	Rio	Dezembro	42.0	2015-01-01
5019	2016	Rio	Dezembro	79.0	2016-01-01

717 rows × 5 columns

In [46]:
```
dataset[mask].shape[0]
```

Out[46]:
```
717
```

In [60]:
```
#import matplotlib as plt
#dataset["state"].value_counts().plot(kind='barh')
```

In [67]:
```
# fetch data of rio fire only of years greater than 2010
mask1=dataset["state"]=="Rio"
mask2=dataset["year"]>2010
```

In [69]:
```
dataset[mask1 & mask2]
```

Out[69]:

	year	state	month	number	date
4316	2011	Rio	Janeiro	10.0	2011-01-01
4317	2012	Rio	Janeiro	12.0	2012-01-01
4318	2013	Rio	Janeiro	9.0	2013-01-01
4319	2014	Rio	Janeiro	35.0	2014-01-01
4320	2015	Rio	Janeiro	97.0	2015-01-01
...	...	...	...	...	...
5015	2012	Rio	Dezembro	38.0	2012-01-01
5016	2013	Rio	Dezembro	62.0	2013-01-01
5017	2014	Rio	Dezembro	31.0	2014-01-01
5018	2015	Rio	Dezembro	42.0	2015-01-01
5019	2016	Rio	Dezembro	79.0	2016-01-01

249 rows × 5 columns

In [72]:
```
dataset.sort_values("year",ascending=False)
```

Out[72]:

	year	state	month	number	date
3227	2017	Pará	Junho	679.000	2017-01-01
3028	2017	Minas Gerais	Agosto	2.142	2017-01-01
3068	2017	Minas Gerais	Outubro	3.062	2017-01-01
3088	2017	Minas Gerais	Novembro	136.000	2017-01-01
339	2017	Alagoas	Maio	1.000	2017-01-01
...	...	...	...	...	...
3626	1998	Paraiba	Março	0.000	1998-01-01
340	1998	Alagoas	Junho	0.000	1998-01-01
2690	1998	Mato Grosso	Abril	0.000	1998-01-01
6036	1998	Sergipe	Abril	0.000	1998-01-01
0	1998	Acre	Janeiro	0.000	1998-01-01

6454 rows × 5 columns

In [50]:
```
#check are there any null values
dataset.isnull().sum()
```

Out[50]:
```
year      0
state     0
month     0
number    0
date      0
dtype: int64
```

In [51]:
```
dataset.head()
```

Out[51]:

	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

In [62]:

```
#total number of fires reported in Amazonas
dataset[dataset["state"]=="Amazonas"].loc[:,"year":"month"]
```

Out[62]:

	year	state	month
718	1998	Amazonas	Janeiro
719	1999	Amazonas	Janeiro
720	2000	Amazonas	Janeiro
721	2001	Amazonas	Janeiro
722	2002	Amazonas	Janeiro
...	...	...	...
952	2012	Amazonas	Dezembro
953	2013	Amazonas	Dezembro
954	2014	Amazonas	Dezembro
955	2015	Amazonas	Dezembro
956	2016	Amazonas	Dezembro

239 rows × 3 columns

In [68]:

```
dataset.year.unique()
```

Out[68]:

```
array([1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008,
       2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017], dtype=int64)
```

In [69]:

```
dataset.state.unique()
```

Out[69]:

```
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)
```

In [61]:

```
dataset[dataset["state"]=="Amazonas"]["number"].sum()
```

Out[61]:

```
30650.129
```

In [76]:

```
#total number of fires reported in Amazonas year wise
amazon_data=dataset[(dataset["state"]=="Amazonas")]
amazon_data_year=amazon_data.groupby("year")["number"].sum().reset_index()
amazon_data_year
```

Out[76]:

	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

In [77]:

```
amazon_data.head(3)
```

Out[77]:

	year	state	month	number	date
718	1998	Amazonas	Janeiro	0.0	1998-01-01
719	1999	Amazonas	Janeiro	3.0	1999-01-01
720	2000	Amazonas	Janeiro	7.0	2000-01-01

In [28]:

```
#plt.plot(amazon_data["year"], amazon_data["number"])

#plt.show()

#print("Generally, a quantity that increase very quickly in the begining, a
```

Rename months name to English

In [15]:
```
dataset.month.unique()
```

Out[15]:
```
array(['Janeiro', 'Fevereiro', 'Março', 'Abril', 'Maio', 'Junho', 'Julho',
       'Agosto', 'Setembro', 'Outubro', 'Novembro', 'Dezembro'],
      dtype=object)
```

In [2]:
```
# Create a Pandas Series
data = {'numbers': [1, 2, 3, 4, 5]}
df = pd.DataFrame(data)

# Define a function to double a number
def double_number(x):
    return x * 2

# Apply the function to the 'numbers' column using .apply()
df['doubled'] = df['numbers'].map(double_number)

print(df)
```

```
   numbers  doubled
0        1        2
1        2        4
2        3        6
3        4        8
4        5       10
```

In [6]:
```
dataset['month_new']=dataset['month'].map({'Janeiro':'jan',
                                        'Fevereiro':'feb',
                                        'Março':'march',
                                        'Abril':'april',
                                        'Maio':'may',
                                        'Junho':'june',
                                        'Julho':'july',
                                        'Agosto':'august',
                                        'Setembro':'september',
                                        'Outubro':'october',
                                        'Novembro':'november',
                                        'Dezembro':'december'

                                       })
```

In [7]:
```
dataset.head()
```

Out[7]:

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

In [10]:
```
dataset.columns
```

Out[10]:
```
Index(['year', 'state', 'month', 'number', 'date', 'month_new'], dtype='ob
ject')
```

In [36]:
```
# total number of fires month wise
data1=dataset.groupby('month_new')['number'].sum().reset_index()
data1
```

Out[36]:

	month_new	number
0	april	28188.770
1	august	88050.435
2	december	57535.480
3	feb	30848.050
4	jan	47747.844
5	july	92326.113
6	june	56010.675
7	march	30717.405
8	may	34731.363
9	november	85508.054
10	october	88681.579
11	september	58578.305

In [38]:
```
# total number of fires month wise
data1=dataset.groupby('month_new')['number'].sum().reset_index()
data1
```

Out[38]:

	month_new	number
0	april	28188.770
1	august	88050.435
2	december	57535.480
3	feb	30848.050
4	jan	47747.844
5	july	92326.113
6	june	56010.675
7	march	30717.405
8	may	34731.363
9	november	85508.054
10	october	88681.579
11	september	58578.305

In [ ]:

```
# total number of fires month wise
data1=dataset.groupby('month_new')['number'].sum().reset_index()
data1
```

In [41]:

```
# multiple aggregate functions with groupby
data2=dataset.groupby('month_new').agg({'number':['mean','max','count','sum
data2
```

Out[41]:

	number				
	mean	max	count	sum	min
month_new					
april	52.201426	947.0	540	28188.770	0.0
august	163.056361	995.0	540	88050.435	0.0
december	112.154932	956.0	513	57535.480	0.0
feb	57.126019	871.0	540	30848.050	0.0
jan	88.258492	960.0	541	47747.844	0.0
july	170.974283	989.0	540	92326.113	0.0
june	103.723472	979.0	540	56010.675	0.0
march	56.884083	820.0	540	30717.405	0.0
may	64.317339	942.0	540	34731.363	0.0
november	158.348248	995.0	540	85508.054	0.0
october	164.225146	964.0	540	88681.579	0.0
september	108.478343	998.0	540	58578.305	0.0

TASKS

1. In which year max no of fires were reported
2. Find average number of fires reported from highest to lowest with reference to state
3. Find the state names where fire was reported in Dec
4. Report top 3 states where highest number of fires were reported.
5. Report fires from Bahia, Acre, and Rio fetch data from 2010 to 2015 and number of fires greater than 0.
6. Report year wise fires of the state with highest number of fires
7. Find aggregate(sum,count, avg, max, min) of number of fires state wise