```
In [136]: se
```

```
Out[136]: array([array([[2, 4],
                 [6, 8]]), 6, 14], dtype=object)
```

```
In [157]: newMatx=np.array([[r1,r2],[8,5]])
```

```
In [158]: newMatx
```

```
Out[158]: array([[ 6, 14],
                 [ 8,  5]])
```

```
In [159]: np.concatenate((se,newMatx))
```

```
Out[159]: array([[ 2,  4],
                 [ 6,  8],
                 [ 6, 14],
                 [ 8,  5]])
```

Pandas

- Usecases
    - Data Trasformation
    - Data visuvalization
    - Data Cleaning
- Natations
    - Series Data Set
    - DataFrames

In [1]:

```python
import pandas as pd

print(dir(pd))
```

```
['Categorical', 'CategoricalDtype', 'CategoricalIndex', 'DataFrame', 'DateOffse
t', 'DatetimeIndex', 'DatetimeTZDtype', 'ExcelFile', 'ExcelWriter', 'Float64Ind
ex', 'Grouper', 'HDFStore', 'Index', 'IndexSlice', 'Int16Dtype', 'Int32Dtype',
'Int64Dtype', 'Int64Index', 'Int8Dtype', 'Interval', 'IntervalDtype', 'Interval
Index', 'MultiIndex', 'NaT', 'Panel', 'Period', 'PeriodDtype', 'PeriodIndex',
'RangeIndex', 'Series', 'SparseArray', 'SparseDataFrame', 'SparseDtype', 'Spars
eSeries', 'TimeGrouper', 'Timedelta', 'TimedeltaIndex', 'Timestamp', 'UInt16Dty
pe', 'UInt32Dtype', 'UInt64Dtype', 'UInt64Index', 'UInt8Dtype', '__builtins__',
'__cached__', '__doc__', '__docformat__', '__file__', '__git_version__', '__loa
der__', '__name__', '__package__', '__path__', '__spec__', '__version__', '_has
htable', '_lib', '_libs', '_np_version_under1p13', '_np_version_under1p14', '_n
p_version_under1p15', '_np_version_under1p16', '_np_version_under1p17', '_tsli
b', '_version', 'api', 'array', 'arrays', 'bdate_range', 'compat', 'concat', 'c
ore', 'crosstab', 'cut', 'date_range', 'datetime', 'describe_option', 'errors',
'eval', 'factorize', 'get_dummies', 'get_option', 'infer_freq', 'interval_rang
e', 'io', 'isna', 'isnull', 'lreshape', 'melt', 'merge', 'merge_asof', 'merge_o
rdered', 'notna', 'notnull', 'np', 'offsets', 'option_context', 'options', 'pan
das', 'period_range', 'pivot', 'pivot_table', 'plotting', 'qcut', 'read_clipboa
rd', 'read_csv', 'read_excel', 'read_feather', 'read_fwf', 'read_gbq', 'read_hd
f', 'read_html', 'read_json', 'read_msgpack', 'read_parquet', 'read_pickle', 'r
ead_sas', 'read_sql', 'read_sql_query', 'read_sql_table', 'read_stata', 'read_t
able', 'reset_option', 'set_eng_float_format', 'set_option', 'show_versions',
'test', 'testing', 'timedelta_range', 'to_datetime', 'to_msgpack', 'to_numeri
c', 'to_pickle', 'to_timedelta', 'tseries', 'unique', 'util', 'value_counts',
'wide_to_long']
```

In [27]:

```python
# Series Data Notations
fmid={'psa':24,'dsp':12,'emf':9,'m4':30,"psd":28,"ds":2}
type(fmid)
```

Out[27]:

```
dict
```

In [3]:

```python
#Series
fmid=pd.Series(fmid)
fmid
```

Out[3]:

```
psa    24
dsp    12
emf     9
m4     30
psd    28
ds      2
dtype: int64
```

In [4]:

```python
smid={'psa':30,'dsp':29,'emf':30,'m4':4,"psd":18,"ds":30}
smid=pd.Series(smid)
smid
```

Out[4]:

```
psa    30
dsp    29
emf    30
m4      4
psd    18
ds     30
dtype: int64
```

In [5]:

```python
df={"First Mid marks ":fmid,"Second Mid Marks ":smid}
df=pd.DataFrame(df)
print(df)
df.columns[1]

```

```
     First Mid marks   Second Mid Marks 
psa               24                  30
dsp               12                  29
emf                9                  30
m4                30                   4
psd               28                  18
ds                 2                  30
```

Out[5]:

```
'Second Mid Marks '
```

In [6]:

```python
df.values
```

Out[6]:

```
array([[24, 30],
       [12, 29],
       [ 9, 30],
       [30,  4],
       [28, 18],
       [ 2, 30]], dtype=int64)
```

In [7]:

```python
df.values[0][0]
```

Out[7]:

```
24
```

In [8]:

```python
#sum of fmid + smid marks
df.values[0][0]+df.values[0][1]
```

Out[8]:

```
54
```

In [12]:

```python
total_marks_list=[]
for i in range(len(df.values)):
    total_marks_list.append(df.values[i][0]+df.values[i][1])
total_marks_list

```

Out[12]:

```
[54, 41, 39, 34, 46, 32]
```

In [14]:

```python
df['Total Marks ']=total_marks_list
```

In [15]:

```
df
```

Out[15]:

	First Mid marks	Second Mid Marks	Total Marks
psa	24	30	54
dsp	12	29	41
emf	9	30	39
m4	30	4	34
psd	28	18	46
ds	2	30	32

In [28]:

```
fmid
```

Out[28]:

```
{'psa': 24, 'dsp': 12, 'emf': 9, 'm4': 30, 'psd': 28, 'ds': 2}
```

In [32]:

```
subjects=list(fmid.keys())
subjects
```

Out[32]:

```
['psa', 'dsp', 'emf', 'm4', 'psd', 'ds']
```

In [33]:

```
total_marks_list
```

Out[33]:

```
[54, 41, 39, 34, 46, 32]
```

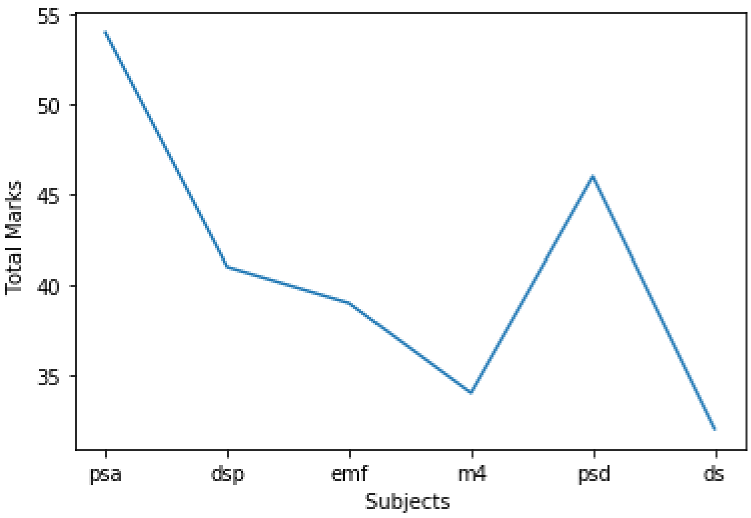
In [42]:

```
import matplotlib.pyplot as plt

plt.plot(subjects,total_marks_list)
plt.xlabel("Subjects ")
plt.ylabel("Total Marks ")
plt.show
```

Out[42]:

```
<function matplotlib.pyplot.show(*args, **kw)>
```



In [45]:

```python
filepath='Income (3).csv'
import pandas as pd
fd=pd.read_csv(filepath)
fd
```

Out[45]:

	GEOID	State	2005	2006	2007	2008	2009	2010	2011	2012	2013
0	04000US01	Alabama	37150	37952	42212	44476	39980	40933	42590	43464	41381
1	04000US02	Alaska	55891	56418	62993	63989	61604	57848	57431	63648	61137
2	04000US04	Arizona	45245	46657	62993	46914	45739	46896	48621	47044	50602
3	04000US05	Arkansas	36658	37057	40795	39586	36538	38587	41302	39018	39919
4	04000US06	California	51755	55319	55734	57014	56134	54283	53367	57020	57528

In [46]:

```python
type(fd)
```

Out[46]:

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

In [48]:

```python
fd.columns
```

Out[48]:

```
Index(['GEOID', 'State', '2005', '2006', '2007', '2008', '2009', '2010',
       '2011', '2012', '2013'],
      dtype='object')
```

In [56]:

```python
fd.columns[9]

```

Out[56]:

```
'2012'
```

In [59]:

```python
fd.values
```

Out[59]:

```
array([['04000US01', 'Alabama', 37150, 37952, 42212, 44476, 39980, 40933,
        42590, 43464, 41381],
       ['04000US02', 'Alaska', 55891, 56418, 62993, 63989, 61604, 57848,
        57431, 63648, 61137],
       ['04000US04', 'Arizona', 45245, 46657, 62993, 46914, 45739, 46896,
        48621, 47044, 50602],
       ['04000US05', 'Arkansas', 36658, 37057, 40795, 39586, 36538,
        38587, 41302, 39018, 39919],
       ['04000US06', 'California', 51755, 55319, 55734, 57014, 56134,
        54283, 53367, 57020, 57528]], dtype=object)
```

In [60]:

```python
fd.values[0]
```

Out[60]:

```
array(['04000US01', 'Alabama', 37150, 37952, 42212, 44476, 39980, 40933,
       42590, 43464, 41381], dtype=object)
```

In [71]:
```
#Sum of Alabama income
total_income_alabama=fd.values[0][2:]
type(total_income_alabama)
total_income_alabama=list(total_income_alabama)
total_income_alabama
alabamafinal_income=sum(total_income_alabama)
alabamafinal_income
```

Out[71]:
```
370138
```

In [72]:
```
fd.describe()
```

Out[72]:

	2005	2006	2007	2008	2009	2010	
count	5.000000	5.000000	5.000000	5.000000	5.00000	5.000000	5.0
mean	45339.800000	46680.600000	52945.400000	50395.800000	47999.00000	47709.400000	48662.2
std	8586.880615	9195.672966	10868.796129	9906.997487	10630.62477	8303.656201	6893.3
min	36658.000000	37057.000000	40795.000000	39586.000000	36538.00000	38587.000000	41302.0
25%	37150.000000	37952.000000	42212.000000	44476.000000	39980.00000	40933.000000	42590.0
50%	45245.000000	46657.000000	55734.000000	46914.000000	45739.00000	46896.000000	48621.0
75%	51755.000000	55319.000000	62993.000000	57014.000000	56134.00000	54283.000000	53367.0
max	55891.000000	56418.000000	62993.000000	63989.000000	61604.00000	57848.000000	57431.0

In [ ]:
```

```

In [61]:
```
fd.values[0][1]
```

Out[61]:
```
'Alabama'
```

In [63]:
```
#Country Names
for i in range(len(fd.values)):
    print(fd.values[i][1])
```

```
Alabama
Alaska
Arizona
Arkansas
California
```

In [73]:
```
years=list(fd.columns[2:])
years
```

Out[73]:
```
['2005', '2006', '2007', '2008', '2009', '2010', '2011', '2012', '2013']
```

In [75]:
```
fd.values[0][1]
```

Out[75]:
```
'Alabama'
```

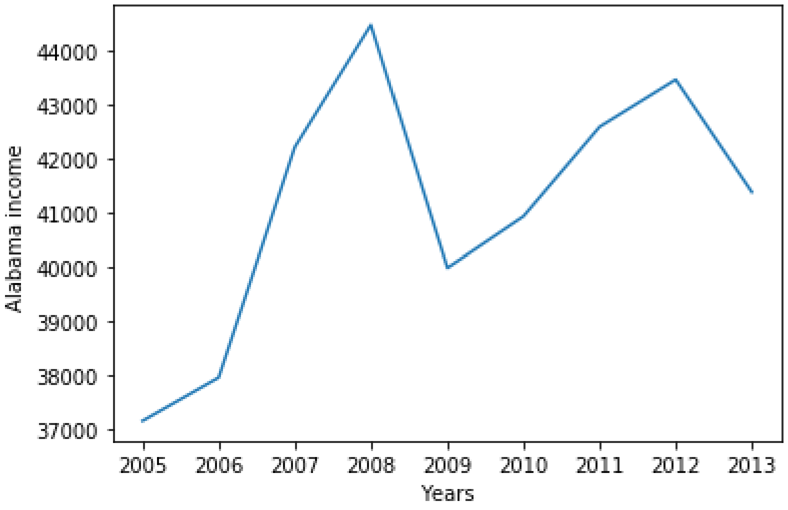
In [76]:
```
total_income_alabama
```

Out[76]:
```
[37150, 37952, 42212, 44476, 39980, 40933, 42590, 43464, 41381]
```

In [78]:
```
import matplotlib.pyplot as plt

plt.plot(years,total_income_alabama)
plt.xlabel("Years")
plt.ylabel("Alabama income ")
plt.show
```

Out[78]:
```
<function matplotlib.pyplot.show(*args, **kw)>
```



In [79]:
```
path='Energy Indicators.xls'
energy=pd.read_excel(path)
energy
```

257	NaN	7	Data for kerosene-type jet fuel include other ...	NaN	NaN
258	NaN	8	Data include Timor-Leste until 2001.	NaN	NaN
259	NaN	9	Data include San Marino and the Holy See.	NaN	NaN
260	NaN	10	Data include Okinawa.	NaN	NaN
261	NaN	11	The data for crude oil production include 50 p...	NaN	NaN
			Data exclude Suriname and		

In [82]:
```
energy.head()
```

Out[82]:

	Unnamed: 0	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5
0	NaN	NaN	Environmental Indicators: Energy	NaN	NaN	NaN
1	NaN	NaN	NaN	NaN	NaN	NaN
2	NaN	NaN	Energy Supply and Renewable Electricity Produc...	NaN	NaN	NaN
3	NaN	NaN	NaN	NaN	NaN	NaN
4	NaN	NaN	NaN	NaN	NaN	Last update: December 2015

In [90]:
```
energy=pd.read_excel(path,skiprows=17,skipfooter=245)
```

In [88]:
```
energy
```

[illegible]	NaN	Angola	Angola	[illegible]	[illegible]	[illegible]
6	NaN	Anguilla	Anguilla	2	136	0.000000
7	NaN	Antigua and Barbuda	Antigua and Barbuda	8	84	0.000000
8	NaN	Argentina	Argentina	3378	79	24.064520
9	NaN	Armenia	Armenia	143	48	28.236060
10	NaN	Aruba	Aruba	12	120	14.870690
11	NaN	Australia	Australia1	5386	231	11.810810
12	NaN	Austria	Austria	1391	164	72.452820
13	NaN	Azerbaijan	Azerbaijan	567	60	6.384345
14	NaN	Bahamas	Bahamas	45	118	0.000000
15	NaN	Bahrain	Bahrain	574	425	0.000000
16	NaN	Bangladesh	Bangladesh	1625	10	1.966329
17	NaN	Barbados	Barbados	19	69	0.000000
18	NaN	Belarus	Belarus	1142	120	0.463389

In [ ]:
```

```