Python Library for Data Analysis ¶

Pandas -Data Analysis

- Series Object -one-dimensional
- Processing/Accessing DataFrames
  - Columns
  - Rows
-

In [3]:

```
# Function to read csv data into a Data Frame
# retruns the DataFrame Object

import pandas as pd
  # comma seprated values all spreads are csv files
def readCSVdata(filepath):
    return pd.read_csv(filepath)
filepath='DataFiles\income.csv'
readCSVdata(filepath)
```

Out[3]:

	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	47215	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 [146]:

```
incomedf.values[3][1]
```

Out[146]:

```
'Arkansas'
```

In [ ]:

```
Print the names of states based on descending order of income in the year 2009
```

In [149]:

```
def avg(df):
    for i in range(len(df.values)):
        s=0
        c=0
        for j in range(2,len(df.columns)):
            s=s+df.values[i][j]
            c=c+1
        print(df.values[i][1],':',s//c,end=" ")
        print('\n')
avg(incomedf)
```

```
Alabama : 41126

Alaska : 60106

Arizona : 48967

Arkansas : 38828

California : 55350
```

In [198]:

```
# Print the names of states based on descending order of income in the year 2009
def desc_2009(df):
    u=[]
    li=[]
    s=[]
    for i in range(len(df.values)):
        for j in range(len(df.columns)):
            a=df.values[i][6]
            if a not in li:
                li.append(a)
    u=sorted(li,reverse=True)
    print(u)
    #for k in range(len(df.values)):
        #print(df.values[k][1],':',u[k])

desc_2009(incomedf)
```

```
[61604, 56134, 45739, 39980, 36538]
```

In [4]:

```
incomedf=readCSVdata(filepath)
# Fucntion to print all columns names in a single line

# GEOID State 2005 2006 2007 2008 2009 2010 2011 2012 2013

def printDataFrameColumns(df):
    columns=df.columns
    for column in columns:
        print(column,end=" ")
    return
printDataFrameColumns(incomedf)
```

```
GEOID State 2005 2006 2007 2008 2009 2010 2011 2012 2013
```

In [7]:

```python
# Function to access a row based on a unique
def accessDataFrameRow(df,key):
    for row in df.values:
        if key in row:
            for item in row:
                print(item,end=" ")
    return
accessDataFrameRow(incomedf,'Alaska')
```

```
04000US02 Alaska 55891 56418 62993 63989 61604 57848 57431 63648 61137
```

In [8]:

```python
incomedf.values[2][4]=62994
incomedf
```

Out[8]:

	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	47215	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 [94]:

```python
def accessDataFromRow(df,key):
    for row in df.values:
        if key in row:
            for item in row:
                print(item,end=" ")
            print('\n')
    return
accessDataFromRow(incomedf,'Alaska')
```

```
04000US02 Alaska 55891 56418 62993 63989 61604 57848 57431 63648 61137
```

In [5]:

```python
incomedf.values[0][2]
#incomedf
```

Out[5]:

```
37150
```

In [56]:

```
# Accessing a unique value based on row,column
# Income of a state in a given year
def getRowIndex(df,rowkey):
    for i in range(len(df.values)):
        if df.values[i][0]==rowkey or df.values[i][1]==rowkey:
            rowindex=i
    return rowindex
def getColumnIndex(df,columnkey):
    for i in range(len(df.columns)):
        if df.columns[i]==columnkey:
            columnindex=i
    return columnindex

def valueFromRowColumn(df,rowkey,columnkey):
    for i in range(len(df .values)):
        if df.values[i][0]==rowkey or df.values[i][1]==rowkey:
            rowindex=i
    for i in range(len(df.columns)):
        if df.columns[i]==columnkey:
            columnindex=i
    return df.values[rowindex][columnindex]
valueFromRowColumn(incomedf,'California','2010')
```

Out[56]:

```
54283
```

In [11]:

```
a=incomedf.values[0][1]

a
```

Out[11]:

```
'Alabama'
```

In [85]:

```
def getRowIndex(df,rowkey):
    for i in range(len(df.values)):
        if df.values[i][0]==rowkey or df.values[i][1]==rowkey:
            rowindex=i
    return rowindex
getRowIndex(incomedf,'Arizona')
```

Out[85]:

```
2
```

In [86]:

```
def getColumnIndex(df,columnkey):
    for i in range(len(df.columns)):
        if df.columns[i]==columnkey:
            columnindex=i
    return columnindex
getColumnIndex(incomedf,'2005')
```

Out[86]:

```
2
```

In [15]:

```
def updateDataFromRowColumn(df,rowkey,columnkey,newdata):
    rowindex=getRowIndex(df,rowkey)
    columnindex=getColumnIndex(df,columnkey)
    row=df.values[rowindex]
    row[columnindex]=newdata
    df.loc[rowindex]=row
    return

updateDataFromRowColumn(incomedf,'Arizona','2007',62993)
incomedf
```

Out[15]:

	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 [67]:

```
accessDataFromRow(incomedf, 55734)
```

```
04000US06 California 51755 55319 55734 57014 56134 54283 53367 57020 57528
```

In [18]:

```
# Function to write DataFrame to row
incomedf.to_csv(filepath,index=False)
```

In [22]:

```
# Function to add a new row of data to Data
def addRowDataDataFrame(df,rowdata):
    lastrowindex=len(df.values)-1
    df.loc[lastrowindex+1]=rowdata
    return
rowdata=[1,2,3,4,5,56,13,14,313,3,0]
addRowDataDataFrame(incomedf,rowdata)
```

In [23]:

```
incomedf
```

Out[23]:

	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
5	1	2	3	4	5	56	13	14	313	3	0
6	1	2	3	4	5	56	13	14	313	3	0

In [27]:
```
def deleteRowDataFrame(df,rowkey):
    rowindex=getRowIndex(df,rowkey)
    return df.drop(rowindex)
    incomedf=deleteRowDataFrame(incomedf,1)
incomedf
```

Out[27]:

	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
5	1	2	3	4	5	56	13	14	313	3	0

In [ ]: