In [1]:
```
# Function to read csv data then print it
import pandas as pd
def readCSVfile(filepath):
    return pd.read_csv(filepath)

filepath='DataFiles/Income.csv'
readCSVfile(filepath)
```

Out[1]:

	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 [2]:
```
incomedf = readCSVfile(filepath)

# Function to print all columns
def printDataFrameColumms(df):
    columns=df.columns
    for column in columns:
        print(column,end=' ')
    return

printDataFrameColumms(incomedf)

```

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

In [3]:
```
# Function to access entire row based on a unique value in row
def printUniqueRow(df,key):
    for row in df.values:
        if key in row:
            for item in row:
                print(item,end=' ')
            print('\n')
    return

printUniqueRow(incomedf,'Alaska')
```

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

In [4]:
```
# Accessing a unique value based on row, column info
# 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

getRowIndex(incomedf,'Arizona')
```

Out[4]: 2

In [5]:
```
def getColumnIndex(df,columnkey):
    for i in range(len(df.columns)):
        if df.columns[i]==columnkey:
            columnindex=i
    return columnindex

#getColumnIndex(incomedf,'2006')
```

In [6]:
```
def valueRowColumn(df,rowkey,columnkey):
    rowindex=getRowIndex(df,rowkey)
    columnindex=getColumnIndex(df,columnkey)
    return df.values[rowindex][columnindex]
valueRowColumn(incomedf,'Alaska','2006')

```

Out[6]: 56418

In [215]:
```
# Function to update data based on rowkey aRnd columnkey
def updateDataFromRowColumn(df,rowkey,columnkey,newdata):
    rowindex=getRowIndex(df,rowkey)
    print(rowindex)
    columnindex=getColumnIndex(df,columnkey)
    row=df.values[rowindex]
    row[columnindex]= newdata
    df.loc[rowindex]=row
    return

updateDataFromRowColumn(incomedf,'Arizona','2006',46657)
incomedf

```

```
2
```

Out[215]:

	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 [216]:
```
incomedf.to_csv('DataFiles/updatedincomedata')
# if we write like this index values also included inorder to avoid them
# make index as False i.e
incomedf.to_csv('DataFiles/updatedincomedata',index=False)
```

In [189]:
```
# Function to Add a new row
def addRow(df,newrow):
    lastindex=len(df.values)-1
    df.loc[lastindex+1]=newrow
    return
newdata=[1,2,3,4,5,6,7,8,9,11,2]
# addRow(incomedf,newdata)
incomedf
```

Out[189]:

	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 [190]:
```
# FUnction to delete a row
def deleteRow(df,rowkey):
    rowindex = getRowIndex(df,rowkey)
    return df.drop(rowindex)


# incomedf = deleteRow(incomedf,1)
incomedf
```

Out[190]:

	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

Practice on Income Dataset

- Define functions for the following data points
  - Combined Average Income of all states from 2005 to 2013
  - State with highest average income in the last three years
  - State with lowest average income from 2007 to 2010(inclusive)
  - Print the list of all states in the same line with average income less than California
  - Print the names of states based on descending order of income in the year 2009
  - State with the lowest recorded income from 2005 to 2013

In [217]:
```
# Combined Average Income of all states from 2005 to 2013
def avgIncomeStates(df):
    sum = 0
    count=0
    for row in df.values:
        for i in range(2,len(df.columns)):
            sum +=int(row[i])
            count +=1
    return sum/count
avgIncomeStates(incomedf)
```

Out[217]: 48525.42222222222

In [9]:

```
# Function to State with highest average income in the last three years
# last 3 years average of each state
# compare every state print the highest average state
def highestAvgState(df):
    li = {}
    for row in df.values:
        sum = 0
        count = 0
        for i in range(-1,-4,-1):
            sum +=int(row[i])
            count +=1
        avg=sum/count
        li[row[1]]=avg
    print(li)

    maxAvg=sorted(li.values(),reverse=True)[0]
    for item in li.items():
        if maxAvg == item[1]:
            print(item[0],':',item[1])
    return


highestAvgState(incomedf)
```

```
{'Alabama': 42478.333333333336, 'Alaska': 60738.666666666664, 'Arizona': 48755.
666666666664, 'Arkansas': 40079.666666666664, 'California': 55971.666666666664}
Alaska : 60738.666666666664
```

In [10]:
```
def stateAvgOfLast3years(df):
    li = {}
    val=[]
    for row in df.values:
        sum = 0
        count = 0
        for i in range(-1,-4,-1):
            sum +=int(row[i])
            count +=1
        avg=sum/count
        li[row[1]]=avg
    print(li)
    print()
    val = li.values()
    print(val)
    print()
    maxAvg=max(val)
    for item in li.items():
        if maxAvg == item[1]:
            print("Highest avg state in last 3 years is -",item[0],":",item[


stateAvgOfLast3years(incomedf)
```

```
{'Alabama': 42478.333333333336, 'Alaska': 60738.666666666664, 'Arizona': 48755.
666666666664, 'Arkansas': 40079.666666666664, 'California': 55971.666666666664}

dict_values([42478.333333333336, 60738.666666666664, 48755.666666666664, 40079.
666666666664, 55971.666666666664])

Highest avg state in last 3 years is - Alaska : 60738.666666666664
```

In [134]:
```
incomedf
```

Out[134]:

	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 [11]:
```
# Function to get starting year index
def startIndex(df,startyear):
    for i in range(len(df.columns)):
        if df.columns[i] == startyear:
            startindex = i
    return startindex

#startIndex(incomedf,'2007')

```

In [12]:
```
def endIndex(df,columnkey):
    for i in range(len(df.columns)):
        if df.columns[i]==columnkey:
            endindex=i
    return endindex
#endIndex(incomedf,'2010')
```

In [13]:
```
# Function to State with lowest average income from 2007 to 2010(inclusive)
def StateLowestAvgInRange(df,startyear,endyear):
    startindex=startIndex(df,startyear)
    endindex=endIndex(df,endyear)
    li={}
    lis=[]
    for row in df.values:
        sum = 0
        count = 0
        for i in range(startindex,endindex+1):
            sum +=row[i]
            count +=1
        avg = sum/count
        #print(avg)
        li[row[1]]=avg
        lis.append(avg)
    print(li)
    print(lis)
    for items in li.items():
        if min(lis)==items[1]:
            print("lowest avg state is",items[0],':',items[1])
    return


StateLowestAvgInRange(incomedf,'2007','2010')
```

```
{'Alabama': 41900.25, 'Alaska': 61608.5, 'Arizona': 46691.0, 'Arkansas': 38876.
5, 'California': 55791.25}
[41900.25, 61608.5, 46691.0, 38876.5, 55791.25]
lowest avg state is Arkansas : 38876.5
```

In [219]:
```
#Print the list of all states in the same line with average income less than
def getCaliforniaIndex(df,keyword):
    for i in range(len(df.values)):
        if df.values[i][1]==keyword:
            return i
#getCaliforniaIndex(incomedf,'California')

def avgOfStates(df,keyword):
    dic={}
    li = {}
    caliIndex=getCaliforniaIndex(df,keyword)
    for row in df.values:
        dic[row[1]]=sum(row[2:])/(len(df.columns)-2)
    print(dic)
    for item in dic.items():
        if item[0]=='California':
            avgCal = item[1]
    for item in dic.items():
        if item[1]<avgCal:
            li[item[0]]=item[1]
    return list(sorted(li))



avgOfStates(incomedf,'California')

```

```
{'Alabama': 41126.444444444445, 'Alaska': 60106.555555555555, 'Arizona': 47214.
77777777778, 'Arkansas': 38828.88888888889, 'California': 55350.444444444445}
```

Out[219]:
```
['Alabama', 'Arizona', 'Arkansas']
```

In [182]:
```
incomedf
```

Out[182]:

	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 [220]:
```
for row in incomedf.values:
    print(row[2:])
```

```
[37150 37952 42212 44476 39980 40933 42590 43464 41381]
[55891 56418 62993 63989 61604 57848 57431 63648 61137]
[45245 46657 47215 46914 45739 46896 48621 47044 50602]
[36658 37057 40795 39586 36538 38587 41302 39018 39919]
[51755 55319 55734 57014 56134 54283 53367 57020 57528]
```

In [38]:
```
# Print the names of states based on descending order of income in the year
# Get the column index of the year 2009
def get2009Index(df,keyword):
    for i in range(len(df.columns)):
        if df.columns[i]==keyword:
            return i
#get2009Index(incomedf,'2009')

def desOrderIncome(df,keyword):
    dic = {}
    colindex=get2009Index(df,keyword)
    for row in df.values:
        dic[row[1]]=row[colindex]
    print(dic)
    val = sorted(dic.values(),reverse=True)
    print(val)
    for value in val:
        for item in dic.items():
            if value == item[1]:
                print(item[0],end=' ')
    return


desOrderIncome(incomedf,'2009')
```

```
{'Alabama': 39980, 'Alaska': 61604, 'Arizona': 45739, 'Arkansas': 36538, 'Calif
ornia': 56134}
[61604, 56134, 45739, 39980, 36538]
Alaska California Arizona Alabama Arkansas
```

In [70]:
```
# FUnction to State with the lowest recorded income from 2005 to 2013

def lowestIncomeState(df):
    dic={}

    for row in df.values:
        dic[row[1]]=min(list(row[i] for i in range(2,11)))
    print(dic)
    li=min(dic.values())
    i=[]
#     for item in dic.items():
#         if li == item[1]:
#             print(item[0])

lowestIncomeState(incomedf)
```

```
{'Alabama': 37150, 'Alaska': 55891, 'Arizona': 45245, 'Arkansas': 36538, 'Calif
ornia': 51755}
Arkansas
```

In [ ]:
```

```