In [2]:

```
import pandas as pd
```

In [12]:

```
df=pd.read_csv("bigmac.csv",parse_dates=["Date"])#parsing dates from 1/2016
df.head()
```

Out[12]:

| | Date | Country | Price in US Dollars |
|---|---|---|---|
| **0** | 2016-01-01 | Argentina | 2.39 |
| **1** | 2016-01-01 | Australia | 3.74 |
| **2** | 2016-01-01 | Brazil | 3.35 |
| **3** | 2016-01-01 | Britain | 4.22 |
| **4** | 2016-01-01 | Canada | 4.14 |

In [11]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 652 entries, 0 to 651
Data columns (total 3 columns):
 #   Column               Non-Null Count  Dtype         
---  ------               --------------  -----         
 0   Date                 652 non-null    datetime64[ns]
 1   Country              652 non-null    object        
 2   Price in US Dollars  652 non-null    float64       
dtypes: datetime64[ns](1), float64(1), object(1)
memory usage: 15.4+ KB
```

# Create a Multiindex with set index method

In [26]:

```
df=pd.read_csv("bigmac.csv",parse_dates=["Date"])#parsing dates from 1/2016
df.head()
```

Out[26]:

| | Date | Country | Price in US Dollars |
|---|---|---|---|
| **0** | 2016-01-01 | Argentina | 2.39 |
| **1** | 2016-01-01 | Australia | 3.74 |
| **2** | 2016-01-01 | Brazil | 3.35 |
| **3** | 2016-01-01 | Britain | 4.22 |
| **4** | 2016-01-01 | Canada | 4.14 |

In [20]:

```
df.nunique()
```

Out[20]:

```
Date                   12
Country                58
Price in US Dollars    330
dtype: int64
```

In [29]:

```
df.set_index(keys=["Date","Country"],inplace=True)
```

In [30]:

```
df
```

Out[30]:

| | | Price in US Dollars |
|---|---|---|
| **Date** | **Country** | |
| **2016-01-01** | **Argentina** | 2.39 |
| | **Australia** | 3.74 |
| | **Brazil** | 3.35 |
| | **Britain** | 4.22 |
| | **Canada** | 4.14 |
| **...** | **...** | ... |
| **2010-01-01** | **Turkey** | 3.83 |
| | **UAE** | 2.99 |
| | **Ukraine** | 1.83 |
| | **United States** | 3.58 |
| | **Uruguay** | 3.32 |

652 rows × 1 columns

In [32]:

```
df.sort_index(inplace=True)#ascending L>>H
```

In [33]:

```
df.head()
```

Out[33]:

| | | Price in US Dollars |
|---|---|---|
| **Date** | **Country** | |
| **2010-01-01** | **Argentina** | 1.84 |
| | **Australia** | 3.98 |
| | **Brazil** | 4.76 |
| | **Britain** | 3.67 |
| | **Canada** | 3.97 |

In [44]:

```
df.index
```

Out[44]:

```
MultiIndex([('2010-01-01',      'Argentina'),
            ('2010-01-01',      'Australia'),
            ('2010-01-01',         'Brazil'),
            ('2010-01-01',        'Britain'),
            ('2010-01-01',         'Canada'),
            ('2010-01-01',          'Chile'),
            ('2010-01-01',          'China'),
            ('2010-01-01',       'Colombia'),
            ('2010-01-01',     'Costa Rica'),
            ('2010-01-01', 'Czech Republic'),
            ...
            ('2016-01-01',    'Switzerland'),
            ('2016-01-01',         'Taiwan'),
            ('2016-01-01',       'Thailand'),
            ('2016-01-01',         'Turkey'),
            ('2016-01-01',            'UAE'),
            ('2016-01-01',        'Ukraine'),
            ('2016-01-01',  'United States'),
            ('2016-01-01',        'Uruguay'),
            ('2016-01-01',      'Venezuela'),
            ('2016-01-01',        'Vietnam')],
           names=['Date', 'Country'], length=652)
```

In [50]:

```
df.index.names
```

Out[50]:

```
FrozenList(['Date', 'Country'])
```

In [54]:

```
type(df.index)
```

Out[54]:

```
pandas.core.indexes.multi.MultiIndex
```

In [56]:

```
df.index[0]
```

Out[56]:

```
(Timestamp('2010-01-01 00:00:00'), 'Argentina')
```

# Extract index levels values based on get_level_values()

In [59]:

```
df=pd.read_csv("bigmac.csv",parse_dates=["Date"],index_col=["Date","Country"])#parsing date
df.sort_index(inplace=True)
df.head()
```

Out[59]:

| | | Price in US Dollars |
|---|---|---|
| **Date** | **Country** | |
| **2010-01-01** | **Argentina** | 1.84 |
| | **Australia** | 3.98 |
| | **Brazil** | 4.76 |
| | **Britain** | 3.67 |
| | **Canada** | 3.97 |

In [65]:

```
df.index.get_level_values("Date")
df.index.get_level_values(0)
```

Out[65]:

```
DatetimeIndex(['2010-01-01', '2010-01-01', '2010-01-01', '2010-01-01',
               '2010-01-01', '2010-01-01', '2010-01-01', '2010-01-01',
               '2010-01-01', '2010-01-01',
               ...
               '2016-01-01', '2016-01-01', '2016-01-01', '2016-01-01',
               '2016-01-01', '2016-01-01', '2016-01-01', '2016-01-01',
               '2016-01-01', '2016-01-01'],
              dtype='datetime64[ns]', name='Date', length=652, freq=None)
```

In [66]:

```
df.index.get_level_values("Country")
df.index.get_level_values(1)
```

Out[66]:

```
Index(['Argentina', 'Australia', 'Brazil', 'Britain', 'Canada', 'Chile',
       'China', 'Colombia', 'Costa Rica', 'Czech Republic',
       ...
       'Switzerland', 'Taiwan', 'Thailand', 'Turkey', 'UAE', 'Ukraine',
       'United States', 'Uruguay', 'Venezuela', 'Vietnam'],
      dtype='object', name='Country', length=652)
```

# change index level names with set_names method

In [67]:

```
df=pd.read_csv("bigmac.csv",parse_dates=["Date"],index_col=["Date","Country"])#parsing date
df.sort_index(inplace=True)
df.head()
```

Out[67]:

| | | Price in US Dollars |
|---|---|---|
| **Date** | **Country** | |
| **2010-01-01** | **Argentina** | 1.84 |
| | **Australia** | 3.98 |
| | **Brazil** | 4.76 |
| | **Britain** | 3.67 |
| | **Canada** | 3.97 |

In [71]:

```
df.index.set_names(names=["Day","Location"],inplace=True)#change index col names
```

In [72]:

```
df.head()
```

Out[72]:

| | | Price in US Dollars |
|---|---|---|
| **Day** | **Location** | |
| **2010-01-01** | **Argentina** | 1.84 |
| | **Australia** | 3.98 |
| | **Brazil** | 4.76 |
| | **Britain** | 3.67 |
| | **Canada** | 3.97 |

In [75]:

```
#to change only one index
df.index.set_names(names="Day",level=0)#1
df.index.set_names(names="Date",level="Day",inplace=True)#2
df.head()
```

Out[75]:

| | | Price in US Dollars |
|---|---|---|
| **Date** | **Location** | |
| **2010-01-01** | **Argentina** | 1.84 |
| | **Australia** | 3.98 |
| | **Brazil** | 4.76 |
| | **Britain** | 3.67 |
| | **Canada** | 3.97 |

In [77]:

```
df.index.set_names(names="Place",level=1,inplace=True)#1
df.head()
```

Out[77]:

| | | Price in US Dollars |
|---|---|---|
| **Date** | **Place** | |
| **2010-01-01** | **Argentina** | 1.84 |
| | **Australia** | 3.98 |
| | **Brazil** | 4.76 |
| | **Britain** | 3.67 |
| | **Canada** | 3.97 |

In [82]:

```
df.columns.set_names(names="Price",inplace=True)
```

In [83]:

```
df.head()
```

Out[83]:

| | Price | Price in US Dollars |
|---|---|---|
| Date | Place | |
| 2010-01-01 | Argentina | 1.84 |
| | Australia | 3.98 |
| | Brazil | 4.76 |
| | Britain | 3.67 |
| | Canada | 3.97 |

In [90]:

```
df.rename(columns={"Price":"Price($)"},inplace=True)
```

In [91]:

```
df.head()
```

Out[91]:

| | Price | Price($) |
|---|---|---|
| Date | Place | |
| 2010-01-01 | Argentina | 1.84 |
| | Australia | 3.98 |
| | Brazil | 4.76 |
| | Britain | 3.67 |
| | Canada | 3.97 |

# The sort_index() on mutltiindex DF

In [103]:

```
df=pd.read_csv("bigmac.csv",parse_dates=["Date"],index_col=["Date","Country"])#parsing date
df.head()
```

Out[103]:

| | | Price in US Dollars |
|---|---|---|
| **Date** | **Country** | |
| **2016-01-01** | **Argentina** | 2.39 |
| | **Australia** | 3.74 |
| | **Brazil** | 3.35 |
| | **Britain** | 4.22 |
| | **Canada** | 4.14 |

In [106]:

```
df.sort_index(ascending=[False,True]).head(10)#date-desc,country-asc  #1
```

Out[106]:

| | | Price in US Dollars |
|---|---|---|
| **Date** | **Country** | |
| **2016-01-01** | **Argentina** | 2.39 |
| | **Australia** | 3.74 |
| | **Austria** | 3.76 |
| | **Belgium** | 4.25 |
| | **Brazil** | 3.35 |
| | **Britain** | 4.22 |
| | **Canada** | 4.14 |
| | **Chile** | 2.94 |
| | **China** | 2.68 |
| | **Colombia** | 2.43 |

In [109]:

```
df.sort_index(ascending=[True,False],inplace=True)#date-asc,country-desc  #1
```

In [110]:

```
df.head(10)
```

Out[110]:

| | | Price in US Dollars |
|---|---|---|
| **Date** | **Country** | |
| **2010-01-01** | **Uruguay** | 3.32 |
| | **United States** | 3.58 |
| | **Ukraine** | 1.83 |
| | **UAE** | 2.99 |
| | **Turkey** | 3.83 |
| | **Thailand** | 2.11 |
| | **Taiwan** | 2.36 |
| | **Switzerland** | 6.30 |
| | **Sweden** | 5.51 |
| | **Sri Lanka** | 1.83 |

In [115]:

```
df.sort_index(level=1, ascending=True).head()#sort only country #2
```

Out[115]:

| | | Price in US Dollars |
|---|---|---|
| **Date** | **Country** | |
| **2010-01-01** | **Argentina** | 1.84 |
| **2010-07-01** | **Argentina** | 3.56 |
| **2011-07-01** | **Argentina** | 4.84 |
| **2012-01-01** | **Argentina** | 4.64 |
| **2012-07-01** | **Argentina** | 4.16 |

In [113]:

```
df.head(10)
```

Out[113]:

| | | Price in US Dollars |
|---|---|---|
| **Date** | **Country** | |
| **2010-01-01** | **Uruguay** | 3.32 |
| | **United States** | 3.58 |
| | **Ukraine** | 1.83 |
| | **UAE** | 2.99 |
| | **Turkey** | 3.83 |
| | **Thailand** | 2.11 |
| | **Taiwan** | 2.36 |
| | **Switzerland** | 6.30 |
| | **Sweden** | 5.51 |
| | **Sri Lanka** | 1.83 |

In [116]:

```
df.sort_index(level="Country", ascending=True).head()#sort only country #2
```

Out[116]:

| | | Price in US Dollars |
|---|---|---|
| **Date** | **Country** | |
| **2010-01-01** | **Argentina** | 1.84 |
| **2010-07-01** | **Argentina** | 3.56 |
| **2011-07-01** | **Argentina** | 4.84 |
| **2012-01-01** | **Argentina** | 4.64 |
| **2012-07-01** | **Argentina** | 4.16 |

# EXtract rows from multiindex DF-tuples

In [117]:

```
df=pd.read_csv("bigmac.csv",parse_dates=["Date"],index_col=["Date","Country"])#parsing date
df.sort_index(inplace=True)
df.head()
```

Out[117]:

| | | Price in US Dollars |
|---|---|---|
| **Date** | **Country** | |
| **2010-01-01** | **Argentina** | 1.84 |
| | **Australia** | 3.98 |
| | **Brazil** | 4.76 |
| | **Britain** | 3.67 |
| | **Canada** | 3.97 |

In [120]:

```
df.loc["2010-01-01","Argentina"]
```

Out[120]:

```
Price in US Dollars    1.84
Name: (2010-01-01 00:00:00, Argentina), dtype: float64
```

In [122]:

```
df.loc["2010-01-01"]
```

Out[122]:

| | | Price in US Dollars |
|---|---|---|
| **Date** | **Country** | |
| **2010-01-01** | **Argentina** | 1.84 |
| | **Australia** | 3.98 |
| | **Brazil** | 4.76 |
| | **Britain** | 3.67 |
| | **Canada** | 3.97 |
| | **Chile** | 3.18 |
| | **China** | 1.83 |
| | **Colombia** | 3.91 |
| | **Costa Rica** | 3.52 |
| | **Czech Republic** | 3.71 |
| | **Denmark** | 5.99 |
| | **Egypt** | 2.38 |
| | **Euro area** | 4.84 |
| | **Hong Kong** | 1.91 |
| | **Hungary** | 3.86 |
| | **Indonesia** | 2.24 |
| | **Israel** | 3.99 |
| | **Japan** | 3.50 |
| | **Latvia** | 3.09 |
| | **Lithuania** | 2.87 |
| | **Malaysia** | 2.08 |
| | **Mexico** | 2.50 |
| | **New Zealand** | 3.61 |
| | **Norway** | 7.02 |
| | **Pakistan** | 2.42 |
| | **Peru** | 2.81 |
| | **Philippines** | 2.21 |
| | **Poland** | 2.86 |
| | **Russia** | 2.34 |
| | **Saudi Arabia** | 2.67 |
| | **Singapore** | 3.19 |
| | **South Africa** | 2.46 |
| | **South Korea** | 2.98 |
| | **Sri Lanka** | 1.83 |

| | | Price in US Dollars |
|---|---|---|
| **Date** | **Country** | |
| | **Sweden** | 5.51 |
| | **Switzerland** | 6.30 |
| | **Taiwan** | 2.36 |
| | **Thailand** | 2.11 |
| | **Turkey** | 3.83 |
| | **UAE** | 2.99 |
| | **Ukraine** | 1.83 |
| | **United States** | 3.58 |
| | **Uruguay** | 3.32 |

In [125]:

```
df.loc[("2010-01-01","Argentina")]      #tuple   #1
```

Out[125]:

```
Price in US Dollars    1.84
Name: (2010-01-01 00:00:00, Argentina), dtype: float64
```

In [126]:

```
df.loc[("2010-01-01","Argentina"),["Price in US Dollars"]]#3 --DF  <<-- (row)  [columns]
```

Out[126]:

| | | Price in US Dollars |
|---|---|---|
| **Date** | **Country** | |
| **2010-01-01** | **Argentina** | 1.84 |

In [127]:

```
df.loc[("2010-01-01","Argentina"),"Price in US Dollars"]#2-- MI series
```

Out[127]:

```
Date        Country
2010-01-01  Argentina    1.84
Name: Price in US Dollars, dtype: float64
```

In [131]:

```
df.loc[("2010-01-01")].head() #not a tuple caz no comma ,
```

Out[131]:

| | | Price in US Dollars |
|---|---|---|
| **Date** | **Country** | |
| **2010-01-01** | **Argentina** | 1.84 |
| | **Australia** | 3.98 |
| | **Brazil** | 4.76 |
| | **Britain** | 3.67 |
| | **Canada** | 3.97 |

In [130]:

```
df.loc[("2010-01-01",)].head()  #not a tuple caz no comma ,
```

Out[130]:

| | Price in US Dollars |
|---|---|
| **Country** | |
| **Argentina** | 1.84 |
| **Australia** | 3.98 |
| **Brazil** | 4.76 |
| **Britain** | 3.67 |
| **Canada** | 3.97 |

In [136]:

```
df.iloc[0]     #iloc
```

Out[136]:

```
Price in US Dollars    1.84
Name: (2010-01-01 00:00:00, Argentina), dtype: float64
```

In [134]:

```
df.iloc[4]
```

Out[134]:

```
Price in US Dollars    3.97
Name: (2010-01-01 00:00:00, Canada), dtype: float64
```

In [142]:

```
df.iloc[[2,4,85,80]]
```

Out[142]:

| | | Price in US Dollars |
|---|---|---|
| **Date** | **Country** | |
| **2010-01-01** | **Brazil** | 4.76 |
| | **Canada** | 3.97 |
| **2010-07-01** | **Uruguay** | 3.74 |
| | **Thailand** | 2.17 |

# The transpose method()

In [3]:

```
df=pd.read_csv("bigmac.csv",parse_dates=["Date"],index_col=["Date","Country"])#parsing date
df.sort_index(inplace=True)
df.head()
```

Out[3]:

| | | Price in US Dollars |
|---|---|---|
| **Date** | **Country** | |
| **2010-01-01** | **Argentina** | 1.84 |
| | **Australia** | 3.98 |
| | **Brazil** | 4.76 |
| | **Britain** | 3.67 |
| | **Canada** | 3.97 |

In [5]:

```
df.transpose()
```

Out[5]:

| Date | | | | | | | | | | 2010-01-01 |
|---|---|---|---|---|---|---|---|---|---|---|
| **Country** | **Argentina** | **Australia** | **Brazil** | **Britain** | **Canada** | **Chile** | **China** | **Colombia** | **Costa Rica** | **Czech Republic** |
| **Price in US Dollars** | 1.84 | 3.98 | 4.76 | 3.67 | 3.97 | 3.18 | 1.83 | 3.91 | 3.52 | 3.71 |

1 rows × 652 columns

In [6]:

```
df.head()
```

Out[6]:

| | | Price in US Dollars |
|---|---|---|
| **Date** | **Country** | |
| **2010-01-01** | **Argentina** | 1.84 |
| | **Australia** | 3.98 |
| | **Brazil** | 4.76 |
| | **Britain** | 3.67 |
| | **Canada** | 3.97 |

In [7]:

```
df=df.transpose()
```

In [8]:

```
df
```

Out[8]:

| Date | | | | | | | | | | 2010-01-01 |
|---|---|---|---|---|---|---|---|---|---|---|
| **Country** | **Argentina** | **Australia** | **Brazil** | **Britain** | **Canada** | **Chile** | **China** | **Colombia** | **Costa Rica** | **Czech Republic** |
| **Price in US Dollars** | 1.84 | 3.98 | 4.76 | 3.67 | 3.97 | 3.18 | 1.83 | 3.91 | 3.52 | 3.71 |

1 rows × 652 columns

In [21]:

```
df.loc["Price in US Dollars",("2010-01-01","Chile")]#Series
```

Out[21]:

```
Date        Country
2010-01-01  Chile      3.18
Name: Price in US Dollars, dtype: float64
```

In [23]:

```
df.loc[["Price in US Dollars"],("2010-01-01","Chile")]#Dataframe
```

Out[23]:

| | Date | 2010-01-01 |
|---|---|---|
| | Country | Chile |
| Price in US Dollars | | 3.18 |

In [22]:

```
df.loc["Price in US Dollars",[("2010-01-01","Chile"),("2010-01-01","China")]]#series
```

Out[22]:

```
Date        Country
2010-01-01  Chile      3.18
            China      1.83
Name: Price in US Dollars, dtype: float64
```

In [25]:

```
df.loc[["Price in US Dollars"],[("2010-01-01","Chile"),("2010-01-01","China")]]#dataframe
```

Out[25]:

| | Date | 2010-01-01 | |
|---|---|---|---|
| | Country | Chile | China |
| Price in US Dollars | | 3.18 | 1.83 |

In [28]:

```
df.loc[["Price in US Dollars"],[("2010-01-01","Chile"),("2010-01-01","China")]]#dataframe
```

Out[28]:

| | Date | 2010-01-01 | |
|---|---|---|---|
| | Country | Chile | China |
| Price in US Dollars | | 3.18 | 1.83 |

In [31]:

```
df.loc[("Price in US Dollars",),("2010-01-01","Chile")]#Dataframe
```

Out[31]:

| Date | 2010-01-01 |
|---|---|
| Country | Chile |
| Price in US Dollars | 3.18 |

In [41]:

```
df.loc[("Price in US Dollars",),("2010-01-01",)]#Dataframe
```

Out[41]:

| Date | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Country | Argentina | Australia | Brazil | Britain | Canada | Chile | China | Colombia | Costa Rica | Czech Republic |
| Price in US Dollars | 1.84 | 3.98 | 4.76 | 3.67 | 3.97 | 3.18 | 1.83 | 3.91 | 3.52 | 3.71 |

1 rows × 43 columns

In [43]:

```
df.loc[("Price in US Dollars",),("2010-01-01","Australia"):("2010-01-01","Colombia")]#Slici
```

Out[43]:

| Date | | | | | | | 2010-01-01 |
|---|---|---|---|---|---|---|---|
| Country | Australia | Brazil | Britain | Canada | Chile | China | Colombia |
| Price in US Dollars | 3.98 | 4.76 | 3.67 | 3.97 | 3.18 | 1.83 | 3.91 |

In [57]:

```
df.columns.unique(level=0)
#df.index.nunique()
```

Out[57]:

```
DatetimeIndex(['2010-01-01', '2010-07-01', '2011-07-01', '2012-01-01',
               '2012-07-01', '2013-01-01', '2013-07-01', '2014-01-01',
               '2014-07-01', '2015-01-01', '2015-07-01', '2016-01-01'],
              dtype='datetime64[ns]', name='Date', freq=None)
```

In [59]:

```
df.columns.unique(level=1).nunique()
```

Out[59]:

```
58
```

In [60]:

```
df.columns.unique(level=1)
```

Out[60]:

```
Index(['Argentina', 'Australia', 'Brazil', 'Britain', 'Canada', 'Chile',
       'China', 'Colombia', 'Costa Rica', 'Czech Republic', 'Denmark', 'Egyp
t',
       'Euro area', 'Hong Kong', 'Hungary', 'Indonesia', 'Israel', 'Japan',
       'Latvia', 'Lithuania', 'Malaysia', 'Mexico', 'New Zealand', 'Norway',
       'Pakistan', 'Peru', 'Philippines', 'Poland', 'Russia', 'Saudi Arabi
a',
       'Singapore', 'South Africa', 'South Korea', 'Sri Lanka', 'Sweden',
       'Switzerland', 'Taiwan', 'Thailand', 'Turkey', 'UAE', 'Ukraine',
       'United States', 'Uruguay', 'Austria', 'Belgium', 'Estonia', 'Finlan
d',
       'France', 'Germany', 'Greece', 'India', 'Ireland', 'Italy',
       'Netherlands', 'Portugal', 'Spain', 'Venezuela', 'Vietnam'],
      dtype='object', name='Country')
```

In [67]:

```
df.loc[("Price in US Dollars",),("2010-01-01",):("2010-07-01",)]#Dataframe Slicing
```

Out[67]:

| Date | | | | | | | | | | 2010-01-01 |
|---|---|---|---|---|---|---|---|---|---|---|
| **Country** | **Argentina** | **Australia** | **Brazil** | **Britain** | **Canada** | **Chile** | **China** | **Colombia** | **Costa Rica** | **Czech Republic** |
| **Price in US Dollars** | 1.84 | 3.98 | 4.76 | 3.67 | 3.97 | 3.18 | 1.83 | 3.91 | 3.52 | 3.71 |

1 rows × 86 columns

# The swapLevel Method

In [68]:

```
df=pd.read_csv("bigmac.csv",parse_dates=["Date"],index_col=["Date","Country"])#parsing date
df.sort_index(inplace=True)
df.head()
```

Out[68]:

| | | Price in US Dollars |
|---|---|---|
| **Date** | **Country** | |
| **2010-01-01** | **Argentina** | 1.84 |
| | **Australia** | 3.98 |
| | **Brazil** | 4.76 |
| | **Britain** | 3.67 |
| | **Canada** | 3.97 |

In [72]:

```
df.swaplevel().head()#chnaging outer and inner level indices
```

Out[72]:

| | | Price in US Dollars |
|---|---|---|
| **Country** | **Date** | |
| **Argentina** | **2010-01-01** | 1.84 |
| **Australia** | **2010-01-01** | 3.98 |
| **Brazil** | **2010-01-01** | 4.76 |
| **Britain** | **2010-01-01** | 3.67 |
| **Canada** | **2010-01-01** | 3.97 |

In [78]:

```
df.swaplevel("Date","Country")
df.swaplevel("Country","Date")
df.swaplevel(0,1)
df.swaplevel(1,0).head()
```

Out[78]:

| | | Price in US Dollars |
|---|---|---|
| **Country** | **Date** | |
| **Argentina** | **2010-01-01** | 1.84 |
| **Australia** | **2010-01-01** | 3.98 |
| **Brazil** | **2010-01-01** | 4.76 |
| **Britain** | **2010-01-01** | 3.67 |
| **Canada** | **2010-01-01** | 3.97 |

In [80]:

```
df=df.swaplevel(0,1)
```

In [81]:

```
df.head()
```

Out[81]:

| | | Price in US Dollars |
|---|---|---|
| **Country** | **Date** | |
| **Argentina** | **2010-01-01** | 1.84 |
| **Australia** | **2010-01-01** | 3.98 |
| **Brazil** | **2010-01-01** | 4.76 |
| **Britain** | **2010-01-01** | 3.67 |
| **Canada** | **2010-01-01** | 3.97 |

# The .stack()

In [9]:

```
world=pd.read_csv("worldstats.csv",index_col=["country","year"])
world.head()
```

Out[9]:

| | | Population | GDP |
|---|---|---|---|
| **country** | **year** | | |
| **Arab World** | **2015** | 392022276.0 | 2.530102e+12 |
| | **2014** | 384222592.0 | 2.873600e+12 |
| | **2013** | 376504253.0 | 2.846994e+12 |
| | **2012** | 368802611.0 | 2.773270e+12 |
| | **2011** | 361031820.0 | 2.497945e+12 |

In [26]:

```
world=world.stack().to_frame()###series to datframe  #Stack converts columns to rows
```

In [28]:

```
world.head()
```

Out[28]:

| | | | | 0 |
|---|---|---|---|---|
| **country** | **year** | | | |
| **Arab World** | **2015** | **Population** | **0** | 3.920223e+08 |
| | | **GDP** | **0** | 2.530102e+12 |
| | **2014** | **Population** | **0** | 3.842226e+08 |
| | | **GDP** | **0** | 2.873600e+12 |
| | **2013** | **Population** | **0** | 3.765043e+08 |

In [31]:

```
world.rename(columns={0:"new"},inplace=True)#cxhsning column name 0 to new
```

In [32]:

```
world.head()
```

Out[32]:

| | | | | | new |
|---|---|---|---|---|---|
| **country** | **year** | | | | |
| **Arab World** | **2015** | **Population** | **0** | **new** | 3.920223e+08 |
| | | **GDP** | **0** | **new** | 2.530102e+12 |
| | **2014** | **Population** | **0** | **new** | 3.842226e+08 |
| | | **GDP** | **0** | **new** | 2.873600e+12 |
| | **2013** | **Population** | **0** | **new** | 3.765043e+08 |

# The .unstack() mrhtod

In [33]:

```
world=pd.read_csv("worldstats.csv",index_col=["country","year"])
world.head()
```

Out[33]:

| | | Population | GDP |
|---|---|---|---|
| **country** | **year** | | |
| **Arab World** | **2015** | 392022276.0 | 2.530102e+12 |
| | **2014** | 384222592.0 | 2.873600e+12 |
| | **2013** | 376504253.0 | 2.846994e+12 |
| | **2012** | 368802611.0 | 2.773270e+12 |
| | **2011** | 361031820.0 | 2.497945e+12 |

In [37]:

```
s=world.stack()
s.head()
```

Out[37]:

```
country     year
Arab World  2015  Population    3.920223e+08
                  GDP           2.530102e+12
            2014  Population    3.842226e+08
                  GDP           2.873600e+12
            2013  Population    3.765043e+08
dtype: float64
```

In [51]:

```
s.unstack().unstack().unstack()
```

Out[51]:

```
            year  country
Population  1960  Afghanistan           8.994793e+06
                  Albania                        NaN
                  Algeria               1.112489e+07
                  Andorra                        NaN
                  Angola                         NaN
                                             ...
GDP         2015  West Bank and Gaza    1.267740e+10
                  World                 7.343364e+13
                  Yemen, Rep.                    NaN
                  Zambia                2.120156e+10
                  Zimbabwe              1.389294e+10
Length: 28224, dtype: float64
```

In [53]:

```
s.unstack().unstack().unstack().to_frame().head()#series to frame
```

Out[53]:

| | | | 0 |
|---|---|---|---|
| | year | country | |
| **Population** | **1960** | **Afghanistan** | 8994793.0 |
| | | **Albania** | NaN |
| | | **Algeria** | 11124892.0 |
| | | **Andorra** | NaN |
| | | **Angola** | NaN |

# unstack() part 2

In [55]:

```
world.head()
```

Out[55]:

| | | Population | GDP |
|---|---|---|---|
| country | year | | |
| **Arab World** | **2015** | 392022276.0 | 2.530102e+12 |
| | **2014** | 384222592.0 | 2.873600e+12 |
| | **2013** | 376504253.0 | 2.846994e+12 |
| | **2012** | 368802611.0 | 2.773270e+12 |
| | **2011** | 361031820.0 | 2.497945e+12 |

In [101]:

```
s=world.stack()#three level series
s
```

Out[101]:

```
country     year
Arab World  2015  Population    3.920223e+08
                  GDP           2.530102e+12
            2014  Population    3.842226e+08
                  GDP           2.873600e+12
            2013  Population    3.765043e+08
                                    ...
Zimbabwe    1962  GDP           1.117602e+09
            1961  Population    3.876638e+06
                  GDP           1.096647e+09
            1960  Population    3.752390e+06
                  GDP           1.052990e+09
Length: 22422, dtype: float64
```

In [67]:

```
s.unstack().unstack().head()#no level which is inner level
```

Out[67]:

| year | 1960 | 1961 | 1962 | 1963 | 1964 | 1965 | 196 |
|---|---|---|---|---|---|---|---|
| country | | | | | | | |
| **Afghanistan** | 8994793.0 | 9164945.0 | 9343772.0 | 9531555.0 | 9728645.0 | 9935358.0 | 10148841. |
| **Albania** | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| **Algeria** | 11124892.0 | 11404859.0 | 11690152.0 | 11985130.0 | 12295973.0 | 12626953.0 | 12980269. |
| **Andorra** | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| **Angola** | NaN | NaN | NaN | NaN | NaN | NaN | NaN |

5 rows × 112 columns

In [66]:

```
s.unstack().unstack(0).head()#moving outer layer (level=0) to columns
```

Out[66]:

| country | Afghanistan | Albania | Algeria | Andorra | Angola | Antigua and Barbuda | Arab World | Argentina | Arm |
|---|---|---|---|---|---|---|---|---|---|
| year | | | | | | | | | |
| 1960 | 8994793.0 | NaN | 11124892.0 | NaN | NaN | NaN | NaN | NaN | |
| 1961 | 9164945.0 | NaN | 11404859.0 | NaN | NaN | NaN | NaN | NaN | |
| 1962 | 9343772.0 | NaN | 11690152.0 | NaN | NaN | NaN | NaN | 21287682.0 | |
| 1963 | 9531555.0 | NaN | 11985130.0 | NaN | NaN | NaN | NaN | 21621845.0 | |
| 1964 | 9728645.0 | NaN | 12295973.0 | NaN | NaN | NaN | NaN | 21953926.0 | |

5 rows × 504 columns

In [69]:

```
s.unstack(1).head()#moving outer layer (level=0) to columns
```

Out[69]:

| | year | 1960 | 1961 | 1962 | 1963 | 1964 |
|---|---|---|---|---|---|---|
| country | | | | | | |
| Afghanistan | Population | 8.994793e+06 | 9.164945e+06 | 9.343772e+06 | 9.531555e+06 | 9.728645e+06 |
| | GDP | 5.377778e+08 | 5.488889e+08 | 5.466667e+08 | 7.511112e+08 | 8.000000e+08 |
| Albania | Population | NaN | NaN | NaN | NaN | NaN |
| | GDP | NaN | NaN | NaN | NaN | NaN |
| Algeria | Population | 1.112489e+07 | 1.140486e+07 | 1.169015e+07 | 1.198513e+07 | 1.229597e+07 |

5 rows × 56 columns

In [70]:

```
s.head()
```

Out[70]:

```
country     year
Arab World  2015  Population    3.920223e+08
                  GDP           2.530102e+12
            2014  Population    3.842226e+08
                  GDP           2.873600e+12
            2013  Population    3.765043e+08
dtype: float64
```

In [79]:

```
s.unstack(-1).head()#selecting particular index to move into columns with negative numbers
s.unstack(2).head()
```

Out[79]:

| | | Population | GDP |
|---|---|---|---|
| **country** | **year** | | |
| **Afghanistan** | **1960** | 8994793.0 | 5.377778e+08 |
| | **1961** | 9164945.0 | 5.488889e+08 |
| | **1962** | 9343772.0 | 5.466667e+08 |
| | **1963** | 9531555.0 | 7.511112e+08 |
| | **1964** | 9728645.0 | 8.000000e+08 |

In [80]:

```
s.unstack(-2).head()#1
s.unstack(1).head()
```

Out[80]:

| | year | 1960 | 1961 | 1962 | 1963 | 1964 |
|---|---|---|---|---|---|---|
| **country** | | | | | | |
| **Afghanistan** | **Population** | 8.994793e+06 | 9.164945e+06 | 9.343772e+06 | 9.531555e+06 | 9.728645e+06 |
| | **GDP** | 5.377778e+08 | 5.488889e+08 | 5.466667e+08 | 7.511112e+08 | 8.000000e+08 |
| **Albania** | **Population** | NaN | NaN | NaN | NaN | NaN |
| | **GDP** | NaN | NaN | NaN | NaN | NaN |
| **Algeria** | **Population** | 1.112489e+07 | 1.140486e+07 | 1.169015e+07 | 1.198513e+07 | 1.229597e+07 |

5 rows × 56 columns

In [81]:

```
s.unstack(-3).head()
s.unstack(0).head()
```

Out[81]:

| country | | Afghanistan | Albania | Algeria | Andorra | Angola | Antigua and Barbuda | Arab World | Arg |
|---|---|---|---|---|---|---|---|---|---|
| year | | | | | | | | | |
| 1960 | Population | 8.994793e+06 | NaN | 1.112489e+07 | NaN | NaN | NaN | NaN | |
| | GDP | 5.377778e+08 | NaN | 2.723638e+09 | NaN | NaN | NaN | NaN | |
| 1961 | Population | 9.164945e+06 | NaN | 1.140486e+07 | NaN | NaN | NaN | NaN | |
| | GDP | 5.488889e+08 | NaN | 2.434767e+09 | NaN | NaN | NaN | NaN | |
| 1962 | Population | 9.343772e+06 | NaN | 1.169015e+07 | NaN | NaN | NaN | NaN | 2128 |

5 rows × 252 columns

In [83]:

```
s.head()
```

Out[83]:

```
country     year
Arab World  2015  Population    3.920223e+08
                  GDP           2.530102e+12
            2014  Population    3.842226e+08
                  GDP           2.873600e+12
            2013  Population    3.765043e+08
dtype: float64
```

In [103]:

```
s.unstack("year").head()#2
```

Out[103]:

| | year | 1960 | 1961 | 1962 | 1963 | 1964 |
|---|---|---|---|---|---|---|
| country | | | | | | |
| Afghanistan | Population | 8.994793e+06 | 9.164945e+06 | 9.343772e+06 | 9.531555e+06 | 9.728645e+06 |
| | GDP | 5.377778e+08 | 5.488889e+08 | 5.466667e+08 | 7.511112e+08 | 8.000000e+08 |
| Albania | Population | NaN | NaN | NaN | NaN | NaN |
| | GDP | NaN | NaN | NaN | NaN | NaN |
| Algeria | Population | 1.112489e+07 | 1.140486e+07 | 1.169015e+07 | 1.198513e+07 | 1.229597e+07 |

5 rows × 56 columns

In [102]:

```
s.unstack("country").head()
```

Out[102]:

| | country | Afghanistan | Albania | Algeria | Andorra | Angola | Antigua and Barbuda | Arab World | Arg |
|---|---|---|---|---|---|---|---|---|---|
| year | | | | | | | | | |
| 1960 | Population | 8.994793e+06 | NaN | 1.112489e+07 | NaN | NaN | NaN | NaN | |
| | GDP | 5.377778e+08 | NaN | 2.723638e+09 | NaN | NaN | NaN | NaN | |
| 1961 | Population | 9.164945e+06 | NaN | 1.140486e+07 | NaN | NaN | NaN | NaN | |
| | GDP | 5.488889e+08 | NaN | 2.434767e+09 | NaN | NaN | NaN | NaN | |
| 1962 | Population | 9.343772e+06 | NaN | 1.169015e+07 | NaN | NaN | NaN | NaN | 212 |

5 rows × 252 columns

# unstck() part 3

In [104]:

```
world=pd.read_csv("worldstats.csv",index_col=["country","year"])
s=world.stack()
world.head()
```

Out[104]:

| | | Population | GDP |
|---|---|---|---|
| country | year | | |
| Arab World | 2015 | 392022276.0 | 2.530102e+12 |
| | 2014 | 384222592.0 | 2.873600e+12 |
| | 2013 | 376504253.0 | 2.846994e+12 |
| | 2012 | 368802611.0 | 2.773270e+12 |
| | 2011 | 361031820.0 | 2.497945e+12 |

In [106]:

```
s.head()#converting multiple indices to columns
```

Out[106]:

```
country     year
Arab World  2015  Population    3.920223e+08
                  GDP           2.530102e+12
            2014  Population    3.842226e+08
                  GDP           2.873600e+12
            2013  Population    3.765043e+08
dtype: float64
```

In [109]:

```
s.unstack(level=["year","country"])
s.unstack(level=[1,0])
```

Out[109]:

| year | 2015 | 2014 | 2013 | 2012 | 2011 | 201 |
|---|---|---|---|---|---|---|
| country | Arab World | Arab World | Arab World | Arab World | Arab World | Arab Wor |
| Population | 3.920223e+08 | 3.842226e+08 | 3.765043e+08 | 3.688026e+08 | 3.610318e+08 | 3.531122e+( |
| GDP | 2.530102e+12 | 2.873600e+12 | 2.846994e+12 | 2.773270e+12 | 2.497945e+12 | 2.103825e+ |

2 rows × 11211 columns

In [113]:

```
s.unstack("year",fill_value=0).head()#replace nan with 0
```

Out[113]:

| | year | 1960 | 1961 | 1962 | 1963 | 1964 |
|---|---|---|---|---|---|---|
| country | | | | | | |
| Afghanistan | Population | 8.994793e+06 | 9.164945e+06 | 9.343772e+06 | 9.531555e+06 | 9.728645e+06 |
| | GDP | 5.377778e+08 | 5.488889e+08 | 5.466667e+08 | 7.511112e+08 | 8.000000e+08 |
| Albania | Population | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| | GDP | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| Algeria | Population | 1.112489e+07 | 1.140486e+07 | 1.169015e+07 | 1.198513e+07 | 1.229597e+07 |

5 rows × 56 columns

In [115]:

```
s=s.unstack("year",fill_value=0)#replace nan with 0
```

In [116]:

```
s.head()
```

Out[116]:

| | year | 1960 | 1961 | 1962 | 1963 | 1964 |
|---|---|---|---|---|---|---|
| country | | | | | | |
| Afghanistan | Population | 8.994793e+06 | 9.164945e+06 | 9.343772e+06 | 9.531555e+06 | 9.728645e+06 |
| | GDP | 5.377778e+08 | 5.488889e+08 | 5.466667e+08 | 7.511112e+08 | 8.000000e+08 |
| Albania | Population | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| | GDP | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| Algeria | Population | 1.112489e+07 | 1.140486e+07 | 1.169015e+07 | 1.198513e+07 | 1.229597e+07 |

5 rows × 56 columns

# The pivot method

In [126]:

```
sales=pd.read_csv("salesmen.csv",parse_dates=["Date"])
sales["Salesman"]=sales["Salesman"].astype("category")
sales.head()
```

Out[126]:

| | Date | Salesman | Revenue |
|---|---|---|---|
| 0 | 2016-01-01 | Bob | 7172 |
| 1 | 2016-01-02 | Bob | 6362 |
| 2 | 2016-01-03 | Bob | 5982 |
| 3 | 2016-01-04 | Bob | 7917 |
| 4 | 2016-01-05 | Bob | 7837 |

In [127]:

```
sales.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1830 entries, 0 to 1829
Data columns (total 3 columns):
 #   Column    Non-Null Count  Dtype         
---  ------    --------------  -----         
 0   Date      1830 non-null   datetime64[ns]
 1   Salesman  1830 non-null   category      
 2   Revenue   1830 non-null   int64         
dtypes: category(1), datetime64[ns](1), int64(1)
memory usage: 30.7 KB
```

In [128]:

```
sales["Salesman"].unique()
```

Out[128]:

```
['Bob', 'Ronald', 'Dave', 'Jeb', 'Oscar']
Categories (5, object): ['Bob', 'Dave', 'Jeb', 'Oscar', 'Ronald']
```

In [130]:

```
len(sales["Salesman"])
```

Out[130]:

```
1830
```

In [131]:

```
sales["Salesman"].value_counts()
```

Out[131]:

```
Bob       366
Dave      366
Jeb       366
Oscar     366
Ronald    366
Name: Salesman, dtype: int64
```

In [138]:

```
########index=to make index ,columns = to make column indices,values =to put in intersction
sales.pivot(index="Date",columns="Salesman",values="Revenue").head()
```

Out[138]:

| Salesman | Bob | Dave | Jeb | Oscar | Ronald |
|---|---|---|---|---|---|
| Date | | | | | |
| **2016-01-01** | 7172 | 1864 | 4430 | 5250 | 2639 |
| **2016-01-02** | 6362 | 8278 | 8026 | 8661 | 4951 |
| **2016-01-03** | 5982 | 4226 | 5188 | 7075 | 2703 |
| **2016-01-04** | 7917 | 3868 | 3144 | 2524 | 4258 |
| **2016-01-05** | 7837 | 2287 | 938 | 2793 | 7771 |

# The pivot_table() method

In [142]:

```
foods=pd.read_csv("foods.csv")
foods.head()
```

Out[142]:

| | First Name | Gender | City | Frequency | Item | Spend |
|---|---|---|---|---|---|---|
| **0** | Wanda | Female | Stamford | Weekly | Burger | 15.66 |
| **1** | Eric | Male | Stamford | Daily | Chalupa | 10.56 |
| **2** | Charles | Male | New York | Never | Sushi | 42.14 |
| **3** | Anna | Female | Philadelphia | Once | Ice Cream | 11.01 |
| **4** | Deborah | Female | Philadelphia | Daily | Chalupa | 23.49 |

In [143]:

```
foods.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   First Name  1000 non-null   object 
 1   Gender      1000 non-null   object 
 2   City        1000 non-null   object 
 3   Frequency   1000 non-null   object 
 4   Item        1000 non-null   object 
 5   Spend       1000 non-null   float64
dtypes: float64(1), object(5)
memory usage: 47.0+ KB
```

In [144]:

```
foods.pivot_table(values="Spend",index="Gender",aggfunc="mean")
```

Out[144]:

| | Spend |
|---|---|
| **Gender** | |
| **Female** | 50.709629 |
| **Male** | 49.397623 |

In [146]:

```
foods.pivot_table(values="Spend",index="Item",aggfunc="sum")
```

Out[146]:

| Item | Spend |
|---|---|
| **Burger** | 7765.73 |
| **Burrito** | 8270.44 |
| **Chalupa** | 7644.52 |
| **Donut** | 8758.76 |
| **Ice Cream** | 8886.99 |
| **Sushi** | 8742.93 |

In [147]:

```
foods.pivot_table(values="Spend",index="Gender",aggfunc="sum")
```

Out[147]:

| Gender | Spend |
|---|---|
| **Female** | 25963.33 |
| **Male** | 24106.04 |

In [155]:

```
(foods.pivot_table(values="Spend",index=["Item","Gender","City"],aggfunc="sum"))
```

Out[155]:

| | | | Spend |
|---|---|---|---|
| **Item** | **Gender** | **City** | |
| **Burger** | **Female** | **New York** | 1239.04 |
| | | **Philadelphia** | 1639.24 |
| | | **Stamford** | 1216.02 |
| | **Male** | **New York** | 1294.09 |
| | | **Philadelphia** | 938.18 |
| | | **Stamford** | 1439.16 |
| **Burrito** | **Female** | **New York** | 978.95 |
| | | **Philadelphia** | 1458.76 |
| | | **Stamford** | 1820.11 |
| | **Male** | **New York** | 1399.40 |
| | | **Philadelphia** | 1312.93 |
| | | **Stamford** | 1300.29 |
| **Chalupa** | **Female** | **New York** | 876.58 |
| | | **Philadelphia** | 1673.33 |
| | | **Stamford** | 1602.35 |
| | **Male** | **New York** | 1227.77 |
| | | **Philadelphia** | 1114.23 |
| | | **Stamford** | 1150.26 |
| **Donut** | **Female** | **New York** | 1446.78 |
| | | **Philadelphia** | 1639.26 |
| | | **Stamford** | 1656.96 |
| | **Male** | **New York** | 1345.27 |
| | | **Philadelphia** | 1249.36 |
| | | **Stamford** | 1421.13 |
| **Ice Cream** | **Female** | **New York** | 1521.62 |
| | | **Philadelphia** | 1479.22 |
| | | **Stamford** | 1032.03 |
| | **Male** | **New York** | 1603.63 |
| | | **Philadelphia** | 2191.27 |
| | | **Stamford** | 1059.22 |
| **Sushi** | **Female** | **New York** | 1480.29 |
| | | **Philadelphia** | 1742.88 |
| | | **Stamford** | 1459.91 |
| | **Male** | **New York** | 1396.15 |

| | | | Spend |
|---|---|---|---|
| **Item** | **Gender** | **City** | |
| | | **Philadelphia** | 1395.88 |
| | | **Stamford** | 1267.82 |

In [159]:

```
foods.pivot_table(values="Spend",index=["Gender","Item","City"],aggfunc="sum")
```

Out[159]:

| | | | Spend |
|---|---|---|---|
| **Gender** | **Item** | **City** | |
| **Female** | **Burger** | **New York** | 1239.04 |
| | | **Philadelphia** | 1639.24 |
| | | **Stamford** | 1216.02 |
| | **Burrito** | **New York** | 978.95 |
| | | **Philadelphia** | 1458.76 |
| | | **Stamford** | 1820.11 |
| | **Chalupa** | **New York** | 876.58 |
| | | **Philadelphia** | 1673.33 |
| | | **Stamford** | 1602.35 |
| | **Donut** | **New York** | 1446.78 |
| | | **Philadelphia** | 1639.26 |
| | | **Stamford** | 1656.96 |
| | **Ice Cream** | **New York** | 1521.62 |
| | | **Philadelphia** | 1479.22 |
| | | **Stamford** | 1032.03 |
| | **Sushi** | **New York** | 1480.29 |
| | | **Philadelphia** | 1742.88 |
| | | **Stamford** | 1459.91 |
| **Male** | **Burger** | **New York** | 1294.09 |
| | | **Philadelphia** | 938.18 |
| | | **Stamford** | 1439.16 |
| | **Burrito** | **New York** | 1399.40 |
| | | **Philadelphia** | 1312.93 |
| | | **Stamford** | 1300.29 |
| | **Chalupa** | **New York** | 1227.77 |
| | | **Philadelphia** | 1114.23 |
| | | **Stamford** | 1150.26 |
| | **Donut** | **New York** | 1345.27 |
| | | **Philadelphia** | 1249.36 |
| | | **Stamford** | 1421.13 |
| | **Ice Cream** | **New York** | 1603.63 |
| | | **Philadelphia** | 2191.27 |
| | | **Stamford** | 1059.22 |
| | **Sushi** | **New York** | 1396.15 |

| | | | Spend |
|---|---|---|---|
| **Gender** | **Item** | **City** | |
| | | **Philadelphia** | 1395.88 |
| | | **Stamford** | 1267.82 |

In [169]:

```
foods.pivot_table(values="Spend",index=["Gender","Item"],columns="City",aggfunc="sum").head
```

Out[169]:

| | **City** | **New York** | **Philadelphia** | **Stamford** |
|---|---|---|---|---|
| **Gender** | **Item** | | | |
| **Female** | **Burger** | 1239.04 | 1639.24 | 1216.02 |
| | **Burrito** | 978.95 | 1458.76 | 1820.11 |
| | **Chalupa** | 876.58 | 1673.33 | 1602.35 |
| | **Donut** | 1446.78 | 1639.26 | 1656.96 |
| | **Ice Cream** | 1521.62 | 1479.22 | 1032.03 |

In [171]:

```
foods.pivot_table(values="Spend",index=["Gender","Item"],columns=["Frequency","City"],aggfu
```

Out[171]:

| | **Frequency** | **Daily** | | | **Monthly** | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | **City** | **New York** | **Philadelphia** | **Stamford** | **New York** | **Philadelphia** | **Stamford** | **New York** | **Phil** |
| **Gender** | **Item** | | | | | | | | |
| **Female** | **Burger** | 262.67 | 231.68 | 144.66 | 171.86 | 215.05 | 238.49 | 97.89 | |
| | **Burrito** | 224.45 | 321.57 | 195.63 | 122.74 | 34.28 | 67.94 | 189.73 | |
| | **Chalupa** | 43.19 | 23.49 | 95.70 | 158.37 | 289.96 | 161.98 | 35.15 | |
| | **Donut** | 478.10 | 247.40 | 124.35 | 284.53 | 50.25 | 229.30 | 56.07 | |
| | **Ice Cream** | 262.19 | 177.69 | 92.88 | 92.53 | 74.51 | 125.85 | 206.15 | |
| | **Sushi** | 81.07 | 348.53 | 393.27 | 93.16 | 78.71 | 108.39 | 69.33 | |
| **Male** | **Burger** | 319.46 | 112.70 | 197.72 | 187.29 | 213.14 | 27.16 | 90.32 | |
| | **Burrito** | 236.21 | 165.76 | 276.23 | 147.54 | 89.58 | 119.60 | 86.78 | |
| | **Chalupa** | 54.09 | 274.81 | 192.64 | 333.76 | 90.70 | 343.76 | 199.09 | |
| | **Donut** | 230.00 | 190.71 | 129.42 | 183.73 | 259.29 | 119.53 | 219.63 | |
| | **Ice Cream** | 178.22 | 148.69 | 15.17 | 152.20 | 386.55 | 74.36 | 57.54 | |
| | **Sushi** | 225.57 | 280.93 | NaN | 430.86 | 155.59 | 93.03 | 164.01 | |

12 rows × 24 columns

In [173]:

```
foods.pivot_table(values="Spend",index=["Gender","Item"],columns="City",aggfunc="max")
foods.pivot_table(values="Spend",index=["Gender","Item"],columns="City",aggfunc="max")
```

Out[173]:

| | City | New York | Philadelphia | Stamford |
|---|---|---|---|---|
| **Gender** | **Item** | | | |
| **Female** | **Burger** | 98.96 | 97.79 | 85.06 |
| | **Burrito** | 92.25 | 96.79 | 99.21 |
| | **Chalupa** | 98.43 | 99.29 | 98.78 |
| | **Donut** | 95.63 | 96.52 | 91.75 |
| | **Ice Cream** | 97.83 | 88.14 | 97.44 |
| | **Sushi** | 99.51 | 99.02 | 95.43 |
| **Male** | **Burger** | 90.32 | 99.68 | 97.20 |
| | **Burrito** | 98.04 | 93.27 | 95.07 |
| | **Chalupa** | 96.44 | 98.40 | 99.87 |
| | **Donut** | 86.70 | 93.12 | 99.26 |
| | **Ice Cream** | 97.65 | 99.24 | 99.17 |
| | **Sushi** | 93.85 | 97.12 | 98.48 |

In [177]:

```
#pivot table directly on pandas without double quotes for df name
pd.pivot_table(data=foods,values="Spend",index=["Gender","Item"],columns="City",aggfunc="ma
```

Out[177]:

| | City | New York | Philadelphia | Stamford |
|---|---|---|---|---|
| **Gender** | **Item** | | | |
| **Female** | **Burger** | 98.96 | 97.79 | 85.06 |
| | **Burrito** | 92.25 | 96.79 | 99.21 |
| | **Chalupa** | 98.43 | 99.29 | 98.78 |
| | **Donut** | 95.63 | 96.52 | 91.75 |
| | **Ice Cream** | 97.83 | 88.14 | 97.44 |
| | **Sushi** | 99.51 | 99.02 | 95.43 |
| **Male** | **Burger** | 90.32 | 99.68 | 97.20 |
| | **Burrito** | 98.04 | 93.27 | 95.07 |
| | **Chalupa** | 96.44 | 98.40 | 99.87 |
| | **Donut** | 86.70 | 93.12 | 99.26 |
| | **Ice Cream** | 97.65 | 99.24 | 99.17 |
| | **Sushi** | 93.85 | 97.12 | 98.48 |

# the pd.melt() method

In [179]:

```
#oop to pivot table()
sale=pd.read_csv("quarters.csv")
sale
```

Out[179]:

| | Salesman | Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|---|---|
| **0** | Boris | 602908 | 233879 | 354479 | 32704 |
| **1** | Bob | 43790 | 514863 | 297151 | 544493 |
| **2** | Tommy | 392668 | 113579 | 430882 | 247231 |
| **3** | Travis | 834663 | 266785 | 749238 | 570524 |
| **4** | Donald | 580935 | 411379 | 110390 | 651572 |
| **5** | Ted | 656644 | 70803 | 375948 | 321388 |
| **6** | Jeb | 486141 | 600753 | 742716 | 404995 |
| **7** | Stacy | 479662 | 742806 | 770712 | 2501 |
| **8** | Morgan | 992673 | 879183 | 37945 | 293710 |

In [181]:

```
pd.melt(sale,id_vars="Salesman",)#id vars--> column needs to be kept constant
```

Out[181]:

| | Salesman | variable | value |
|---|---|---|---|
| **0** | Boris | Q1 | 602908 |
| **1** | Bob | Q1 | 43790 |
| **2** | Tommy | Q1 | 392668 |
| **3** | Travis | Q1 | 834663 |
| **4** | Donald | Q1 | 580935 |
| **5** | Ted | Q1 | 656644 |
| **6** | Jeb | Q1 | 486141 |
| **7** | Stacy | Q1 | 479662 |
| **8** | Morgan | Q1 | 992673 |
| **9** | Boris | Q2 | 233879 |
| **10** | Bob | Q2 | 514863 |
| **11** | Tommy | Q2 | 113579 |
| **12** | Travis | Q2 | 266785 |
| **13** | Donald | Q2 | 411379 |
| **14** | Ted | Q2 | 70803 |
| **15** | Jeb | Q2 | 600753 |
| **16** | Stacy | Q2 | 742806 |
| **17** | Morgan | Q2 | 879183 |
| **18** | Boris | Q3 | 354479 |
| **19** | Bob | Q3 | 297151 |
| **20** | Tommy | Q3 | 430882 |
| **21** | Travis | Q3 | 749238 |
| **22** | Donald | Q3 | 110390 |
| **23** | Ted | Q3 | 375948 |
| **24** | Jeb | Q3 | 742716 |
| **25** | Stacy | Q3 | 770712 |
| **26** | Morgan | Q3 | 37945 |
| **27** | Boris | Q4 | 32704 |
| **28** | Bob | Q4 | 544493 |
| **29** | Tommy | Q4 | 247231 |
| **30** | Travis | Q4 | 570524 |
| **31** | Donald | Q4 | 651572 |
| **32** | Ted | Q4 | 321388 |
| **33** | Jeb | Q4 | 404995 |
| **34** | Stacy | Q4 | 2501 |

| | Salesman | variable | value |
|---|---|---|---|
| **35** | Morgan | Q4 | 293710 |

In [182]:

```
pd.melt(sale,id_vars="Salesman",var_name="Quarter")
```

Out[182]:

| | Salesman | Quarter | value |
|---|---|---|---|
| **0** | Boris | Q1 | 602908 |
| **1** | Bob | Q1 | 43790 |
| **2** | Tommy | Q1 | 392668 |
| **3** | Travis | Q1 | 834663 |
| **4** | Donald | Q1 | 580935 |
| **5** | Ted | Q1 | 656644 |
| **6** | Jeb | Q1 | 486141 |
| **7** | Stacy | Q1 | 479662 |
| **8** | Morgan | Q1 | 992673 |
| **9** | Boris | Q2 | 233879 |
| **10** | Bob | Q2 | 514863 |
| **11** | Tommy | Q2 | 113579 |
| **12** | Travis | Q2 | 266785 |
| **13** | Donald | Q2 | 411379 |
| **14** | Ted | Q2 | 70803 |
| **15** | Jeb | Q2 | 600753 |
| **16** | Stacy | Q2 | 742806 |
| **17** | Morgan | Q2 | 879183 |
| **18** | Boris | Q3 | 354479 |
| **19** | Bob | Q3 | 297151 |
| **20** | Tommy | Q3 | 430882 |
| **21** | Travis | Q3 | 749238 |
| **22** | Donald | Q3 | 110390 |
| **23** | Ted | Q3 | 375948 |
| **24** | Jeb | Q3 | 742716 |
| **25** | Stacy | Q3 | 770712 |
| **26** | Morgan | Q3 | 37945 |
| **27** | Boris | Q4 | 32704 |
| **28** | Bob | Q4 | 544493 |
| **29** | Tommy | Q4 | 247231 |
| **30** | Travis | Q4 | 570524 |
| **31** | Donald | Q4 | 651572 |
| **32** | Ted | Q4 | 321388 |
| **33** | Jeb | Q4 | 404995 |
| **34** | Stacy | Q4 | 2501 |

| | Salesman | Quarter | value |
|---|---|---|---|
| **35** | Morgan | Q4 | 293710 |

In [183]:

```
pd.melt(sale,id_vars="Salesman",var_name="Quarter",value_name="Revenue")
```

Out[183]:

| | Salesman | Quarter | Revenue |
|---|---|---|---|
| **0** | Boris | Q1 | 602908 |
| **1** | Bob | Q1 | 43790 |
| **2** | Tommy | Q1 | 392668 |
| **3** | Travis | Q1 | 834663 |
| **4** | Donald | Q1 | 580935 |
| **5** | Ted | Q1 | 656644 |
| **6** | Jeb | Q1 | 486141 |
| **7** | Stacy | Q1 | 479662 |
| **8** | Morgan | Q1 | 992673 |
| **9** | Boris | Q2 | 233879 |
| **10** | Bob | Q2 | 514863 |
| **11** | Tommy | Q2 | 113579 |
| **12** | Travis | Q2 | 266785 |
| **13** | Donald | Q2 | 411379 |
| **14** | Ted | Q2 | 70803 |
| **15** | Jeb | Q2 | 600753 |
| **16** | Stacy | Q2 | 742806 |
| **17** | Morgan | Q2 | 879183 |
| **18** | Boris | Q3 | 354479 |
| **19** | Bob | Q3 | 297151 |
| **20** | Tommy | Q3 | 430882 |
| **21** | Travis | Q3 | 749238 |
| **22** | Donald | Q3 | 110390 |
| **23** | Ted | Q3 | 375948 |
| **24** | Jeb | Q3 | 742716 |
| **25** | Stacy | Q3 | 770712 |
| **26** | Morgan | Q3 | 37945 |
| **27** | Boris | Q4 | 32704 |
| **28** | Bob | Q4 | 544493 |
| **29** | Tommy | Q4 | 247231 |
| **30** | Travis | Q4 | 570524 |
| **31** | Donald | Q4 | 651572 |
| **32** | Ted | Q4 | 321388 |
| **33** | Jeb | Q4 | 404995 |
| **34** | Stacy | Q4 | 2501 |

| | Salesman | Quarter | Revenue |
|---|---|---|---|
| **35** | Morgan | Q4 | 293710 |

```
'''id_vars : tuple, list, or ndarray, optional
    Column(s) to use as identifier variables.
value_vars : tuple, list, or ndarray, optional
    Column(s) to unpivot. If not specified, uses all columns that
    are not set as `id_vars`.
var_name : scalar
    Name to use for the 'variable' column. If None it uses
    ``frame.columns.name`` or 'variable'.
value_name : scalar, default 'value'
    Name to use for the 'value' column.
col_level : int or str, optional
    If columns are a MultiIndex then use this level to melt.
ignore_index : bool, default True
    If True, original index is ignored. If False, the original index is retained.
    Index labels will be repeated as necessary.'''
```

In [11]:

```
import pandas as pd
foods=pd.read_csv("foods.csv")
foods.head()
```

Out[11]:

| | First Name | Gender | City | Frequency | Item | Spend |
|---|---|---|---|---|---|---|
| **0** | Wanda | Female | Stamford | Weekly | Burger | 15.66 |
| **1** | Eric | Male | Stamford | Daily | Chalupa | 10.56 |
| **2** | Charles | Male | New York | Never | Sushi | 42.14 |
| **3** | Anna | Female | Philadelphia | Once | Ice Cream | 11.01 |
| **4** | Deborah | Female | Philadelphia | Daily | Chalupa | 23.49 |

In [12]:

```
pd.melt(foods,id_vars="First Name",var_name="Item").head()
```

Out[12]:

| | First Name | Item | value |
|---|---|---|---|
| **0** | Wanda | Gender | Female |
| **1** | Eric | Gender | Male |
| **2** | Charles | Gender | Male |
| **3** | Anna | Gender | Female |
| **4** | Deborah | Gender | Female |

In [18]:

```
pd.melt(foods,id_vars="First Name",var_name="Items",value_name="value-name").tail()
```

Out[18]:

| | First Name | Items | value-name |
|---|---|---|---|
| **4995** | Donna | Spend | 83.53 |
| **4996** | Albert | Spend | 72.88 |
| **4997** | Jean | Spend | 5.85 |
| **4998** | Jessica | Spend | 43.19 |
| **4999** | Brian | Spend | 28.36 |

In [26]:

```
pd.melt(foods,id_vars="First Name",var_name="Items",value_name="value-name",value_vars="Genc
```

Out[26]:

| | First Name | Items | value-name |
|---|---|---|---|
| **995** | Donna | Gender | Female |
| **996** | Albert | Gender | Male |
| **997** | Jean | Gender | Female |
| **998** | Jessica | Gender | Female |
| **999** | Brian | Gender | Male |

In [28]:

```
pd.melt(foods,id_vars="First Name",var_name="Items",value_name="value-name",value_vars="Gen
```

Out[28]:

| | First Name | Items | value-name |
|---|---|---|---|
| **995** | Donna | Gender | Female |
| **996** | Albert | Gender | Male |
| **997** | Jean | Gender | Female |
| **998** | Jessica | Gender | Female |
| **999** | Brian | Gender | Male |

In [30]:

```
import pandas as pd
fd=pd.read_csv("foods.csv",index_col="First Name")
fd.head()
```

Out[30]:

| First Name | Gender | City | Frequency | Item | Spend |
|---|---|---|---|---|---|
| **Wanda** | Female | Stamford | Weekly | Burger | 15.66 |
| **Eric** | Male | Stamford | Daily | Chalupa | 10.56 |
| **Charles** | Male | New York | Never | Sushi | 42.14 |
| **Anna** | Female | Philadelphia | Once | Ice Cream | 11.01 |
| **Deborah** | Female | Philadelphia | Daily | Chalupa | 23.49 |

In [ ]:

```
pd.melt(fd,id_vars="city")
```

In [34]:

```
pd.melt(fd,id_vars="City",ignore_index=False).head()###ignore index
```

Out[34]:

| First Name | City | variable | value |
|---|---|---|---|
| **Wanda** | Stamford | Gender | Female |
| **Eric** | Stamford | Gender | Male |
| **Charles** | New York | Gender | Male |
| **Anna** | Philadelphia | Gender | Female |
| **Deborah** | Philadelphia | Gender | Female |

In [36]:

```
pd.melt(fd,id_vars="City",ignore_index=True).head()###ignore index
pd.melt(fd,id_vars="City").head()###ignore index
```

Out[36]:

| | City | variable | value |
|---|---|---|---|
| **0** | Stamford | Gender | Female |
| **1** | Stamford | Gender | Male |
| **2** | New York | Gender | Male |
| **3** | Philadelphia | Gender | Female |
| **4** | Philadelphia | Gender | Female |