Manipulation_And_Data_Manipulation

October 9, 2021

[ ]:
```
import numpy as np
import pandas as pd
retail= pd.read_csv('online_retail2.csv')
```

[ ]:
```
retail.head()
```

[ ]:
```
  Invoice StockCode                          Description  Quantity  \
0  489434     85048  15CM CHRISTMAS GLASS BALL 20 LIGHTS        12
1  489434    79323P                   PINK CHERRY LIGHTS        12
2  489434    79323W                  WHITE CHERRY LIGHTS        12
3  489434     22041         RECORD FRAME 7" SINGLE SIZE         48
4  489434     21232       STRAWBERRY CERAMIC TRINKET BOX        24

           InvoiceDate  Price  Customer ID         Country
0  2009-12-01 07:45:00   6.95      13085.0  United Kingdom
1  2009-12-01 07:45:00   6.75      13085.0  United Kingdom
2  2009-12-01 07:45:00   6.75      13085.0  United Kingdom
3  2009-12-01 07:45:00   2.10      13085.0  United Kingdom
4  2009-12-01 07:45:00   1.25      13085.0  United Kingdom
```

[ ]:
```
retail.shape
```

[ ]:
```
(1067371, 8)
```

We have about 1 million rolls

[ ]:
```
# Drop Duplicate rows
retail = retail.drop_duplicates()
```

[ ]:
```
## after removing duplicates
retail.shape
```

[ ]:
```
(1033036, 8)
```

If i have any duplicates in description column

[ ]:
```
retail.Description.isnull()
```

[ ]:
```
0          False
1          False
2          False
3          False
4          False
           …
1067366    False
1067367    False
1067368    False
1067369    False
1067370    False
Name: Description, Length: 1033036, dtype: bool
```

[ ]:
```
### check total null values
retail.Description.isnull().sum()
```

[ ]:
```
4275
```

The null can be in categorial or continues variables, sometimes you get medians

[ ]:
```
#How many values are null in the dataset
retail.isnull().sum().sum()
```

[ ]:
```
239426
```

239,426 entries are null

[ ]:
```
#Drop all the nulls
retail = retail.dropna(axis=0,how='any')
```

[ ]:
```
retail.shape
```

[ ]:
```
(797885, 8)
```

0.1 Conversions

[ ]:
```
retail.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 797885 entries, 0 to 1067370
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   Invoice      797885 non-null  object
 1   StockCode    797885 non-null  object
 2   Description  797885 non-null  object
 3   Quantity     797885 non-null  int64
 4   InvoiceDate  797885 non-null  object
 5   Price        797885 non-null  float64
```

```
 6   Customer ID  797885 non-null  float64
 7   Country      797885 non-null  object
dtypes: float64(2), int64(1), object(5)
memory usage: 54.8+ MB
```

The Invoice_Date is an object, so we need to convert this to date time.

[ ]:
```
retail.InvoiceDate
```

[ ]:
```
0          2009-12-01 07:45:00
1          2009-12-01 07:45:00
2          2009-12-01 07:45:00
3          2009-12-01 07:45:00
4          2009-12-01 07:45:00
                  …
1067366    2011-12-09 12:50:00
1067367    2011-12-09 12:50:00
1067368    2011-12-09 12:50:00
1067369    2011-12-09 12:50:00
1067370    2011-12-09 12:50:00
Name: InvoiceDate, Length: 797885, dtype: object
```

This have a date time but is an object here

[ ]:
```
retail['date']=pd.to_datetime(retail['InvoiceDate'])
```

[ ]:
```
retail['date']
```

[ ]:
```
0         2009-12-01 07:45:00
1         2009-12-01 07:45:00
2         2009-12-01 07:45:00
3         2009-12-01 07:45:00
4         2009-12-01 07:45:00
                  …
1067366   2011-12-09 12:50:00
1067367   2011-12-09 12:50:00
1067368   2011-12-09 12:50:00
1067369   2011-12-09 12:50:00
1067370   2011-12-09 12:50:00
Name: date, Length: 797885, dtype: datetime64[ns]
```

Now I can get the basic feel o my date, like when did the purchase start or end.

[ ]:
```
retail['date'].describe()
```

```
C:\Users\aduzo\Anaconda3\lib\site-packages\ipykernel_launcher.py:1:
FutureWarning: Treating datetime data as categorical rather than numeric in
`.describe` is deprecated and will be removed in a future version of pandas.
Specify `datetime_is_numeric=True` to silence this warning and adopt the future
```

```
behavior now.
  """Entry point for launching an IPython kernel.
```

[ ]:
```
count                  797885
unique                  41439
top       2011-11-14 15:27:00
freq                      543
first     2009-12-01 07:45:00
last      2011-12-09 12:50:00
Name: date, dtype: object
```

[ ]:
```
print(f' first date is: {retail.date.min()}')
print(f' last date is: {retail.date.max()}')
```

```
 first date is: 2009-12-01 07:45:00
 last date is: 2011-12-09 12:50:00
```

0.2 Filtering methods

[ ]:
```
retail.Country.value_counts()
```

[ ]:
```
United Kingdom          716115
Germany                  17339
EIRE                     16014
France                   13897
Netherlands               5137
Spain                     3754
Belgium                   3110
Switzerland               3058
Portugal                  2414
Australia                 1890
Channel Islands           1646
Italy                     1507
Sweden                    1343
Norway                    1308
Cyprus                    1157
Finland                   1049
Austria                    938
Denmark                    797
Greece                     663
Japan                      565
USA                        535
Poland                     527
Unspecified                521
United Arab Emirates       386
Singapore                  346
Israel                     321
Malta                      299
```

```
Canada                    228
Iceland                   222
Lithuania                 154
RSA                       123
Brazil                     94
Thailand                   76
Korea                      63
European Community         61
Bahrain                    59
West Indies                54
Lebanon                    45
Nigeria                    30
Czech Republic             30
Saudi Arabia               10
Name: Country, dtype: int64
```

we see the counts of every country

Display only france

[ ]:
```
# From retail take where retail is equal to France
retail[retail.Country=='France']
```

[ ]:
```
         Invoice StockCode                          Description  Quantity  \
71        489439     22065     CHRISTMAS PUDDING TRINKET POT          12
72        489439     22138      BAKING SET 9 PIECE RETROSPOT           9
73        489439     22139  RETRO SPOT TEA SET CERAMIC 11 PC           9
74        489439     22352   LUNCHBOX WITH CUTLERY RETROSPOT          12
75        489439    85014A   BLACK/BLUE DOTS RUFFLED UMBRELLA          3
…           …        …                                  …        …
1067366   581587     22899       CHILDREN'S APRON DOLLY GIRL           6
1067367   581587     23254      CHILDRENS CUTLERY DOLLY GIRL           4
1067368   581587     23255    CHILDRENS CUTLERY CIRCUS PARADE          4
1067369   581587     22138      BAKING SET 9 PIECE RETROSPOT           3
1067370   581587      POST                             POSTAGE          1

                 InvoiceDate  Price  Customer ID Country                date
71       2009-12-01 09:28:00   1.45      12682.0  France 2009-12-01 09:28:00
72       2009-12-01 09:28:00   4.95      12682.0  France 2009-12-01 09:28:00
73       2009-12-01 09:28:00   4.95      12682.0  France 2009-12-01 09:28:00
74       2009-12-01 09:28:00   2.55      12682.0  France 2009-12-01 09:28:00
75       2009-12-01 09:28:00   5.95      12682.0  France 2009-12-01 09:28:00
…                      …    …           …     …                  …
1067366  2011-12-09 12:50:00   2.10      12680.0  France 2011-12-09 12:50:00
1067367  2011-12-09 12:50:00   4.15      12680.0  France 2011-12-09 12:50:00
1067368  2011-12-09 12:50:00   4.15      12680.0  France 2011-12-09 12:50:00
1067369  2011-12-09 12:50:00   4.95      12680.0  France 2011-12-09 12:50:00
1067370  2011-12-09 12:50:00  18.00      12680.0  France 2011-12-09 12:50:00
```

```
[13897 rows x 9 columns]
```

If both France and EIRE for multiple conditions, we put them in a tuple.

```
[ ]: retail[(retail.Country== 'France')|(retail.Country== 'EIRE')]
```

```
[ ]:         Invoice StockCode                          Description  Quantity  \
71        489439     22065      CHRISTMAS PUDDING TRINKET POT          12
72        489439     22138       BAKING SET 9 PIECE RETROSPOT           9
73        489439     22139  RETRO SPOT TEA SET CERAMIC 11 PC           9
74        489439     22352   LUNCHBOX WITH CUTLERY RETROSPOT          12
75        489439    85014A   BLACK/BLUE DOTS RUFFLED UMBRELLA          3
…          …       …                                     …       …
1067366   581587     22899        CHILDREN'S APRON DOLLY GIRL           6
1067367   581587     23254       CHILDRENS CUTLERY DOLLY GIRL           4
1067368   581587     23255     CHILDRENS CUTLERY CIRCUS PARADE          4
1067369   581587     22138       BAKING SET 9 PIECE RETROSPOT           3
1067370   581587      POST                              POSTAGE          1

                 InvoiceDate  Price  Customer ID Country                date
71       2009-12-01 09:28:00   1.45      12682.0  France 2009-12-01 09:28:00
72       2009-12-01 09:28:00   4.95      12682.0  France 2009-12-01 09:28:00
73       2009-12-01 09:28:00   4.95      12682.0  France 2009-12-01 09:28:00
74       2009-12-01 09:28:00   2.55      12682.0  France 2009-12-01 09:28:00
75       2009-12-01 09:28:00   5.95      12682.0  France 2009-12-01 09:28:00
…                        …    …           …      …                    …
1067366  2011-12-09 12:50:00   2.10      12680.0  France 2011-12-09 12:50:00
1067367  2011-12-09 12:50:00   4.15      12680.0  France 2011-12-09 12:50:00
1067368  2011-12-09 12:50:00   4.15      12680.0  France 2011-12-09 12:50:00
1067369  2011-12-09 12:50:00   4.95      12680.0  France 2011-12-09 12:50:00
1067370  2011-12-09 12:50:00  18.00      12680.0  France 2011-12-09 12:50:00

[29911 rows x 9 columns]
```

```
[ ]: retail[(retail.Country== 'France')|(retail.Country== 'EIRE')].Country.
     ↪value_counts()
```

```
[ ]: EIRE      16014
     France    13897
     Name: Country, dtype: int64
```

Another way of writing it.

```
[ ]: retail[(retail.Country== 'France')|(retail.Country== 'EIRE')]['Country'].
     ↪value_counts()
```

[ ]:
```
EIRE      16014
France    13897
Name: Country, dtype: int64
```

Multiple Countries

[ ]:
```
countries=['France','EIRE','Spain']
```

[ ]:
```
retail[retail.Country.isin(countries)]
```

[ ]:
```
        Invoice StockCode                          Description  Quantity  \
71       489439     22065     CHRISTMAS PUDDING TRINKET POT          12
72       489439     22138      BAKING SET 9 PIECE RETROSPOT           9
73       489439     22139  RETRO SPOT TEA SET CERAMIC 11 PC           9
74       489439     22352   LUNCHBOX WITH CUTLERY RETROSPOT          12
75       489439    85014A   BLACK/BLUE DOTS RUFFLED UMBRELLA          3
…          …        …                                    …        …
1067366  581587     22899       CHILDREN'S APRON DOLLY GIRL           6
1067367  581587     23254      CHILDRENS CUTLERY DOLLY GIRL           4
1067368  581587     23255    CHILDRENS CUTLERY CIRCUS PARADE          4
1067369  581587     22138      BAKING SET 9 PIECE RETROSPOT           3
1067370  581587      POST                              POSTAGE         1

                 InvoiceDate  Price  Customer ID Country                date
71       2009-12-01 09:28:00   1.45      12682.0  France 2009-12-01 09:28:00
72       2009-12-01 09:28:00   4.95      12682.0  France 2009-12-01 09:28:00
73       2009-12-01 09:28:00   4.95      12682.0  France 2009-12-01 09:28:00
74       2009-12-01 09:28:00   2.55      12682.0  France 2009-12-01 09:28:00
75       2009-12-01 09:28:00   5.95      12682.0  France 2009-12-01 09:28:00
…                        …    …           …      …                  …
1067366  2011-12-09 12:50:00   2.10      12680.0  France 2011-12-09 12:50:00
1067367  2011-12-09 12:50:00   4.15      12680.0  France 2011-12-09 12:50:00
1067368  2011-12-09 12:50:00   4.15      12680.0  France 2011-12-09 12:50:00
1067369  2011-12-09 12:50:00   4.95      12680.0  France 2011-12-09 12:50:00
1067370  2011-12-09 12:50:00  18.00      12680.0  France 2011-12-09 12:50:00

[33665 rows x 9 columns]
```

[ ]:
```
retail[retail.Country.isin(countries)].Country.value_counts()
```

[ ]:
```
EIRE      16014
France    13897
Spain      3754
Name: Country, dtype: int64
```

[ ]:
```
retail.columns
```

[ ]:
```
Index(['Invoice', 'StockCode', 'Description', 'Quantity', 'InvoiceDate',
       'Price', 'Customer ID', 'Country', 'date'],
      dtype='object')
```

[ ]:
```
retail[retail.date>='2011']
```

[ ]:
```
        Invoice StockCode                          Description  Quantity  \
567942   539993     22386         JUMBO BAG PINK POLKADOT        10
567943   539993     21499              BLUE POLKADOT WRAP        25
567944   539993     21498             RED RETROSPOT WRAP         25
567945   539993     22379          RECYCLING BAG RETROSPOT          5
567946   539993     20718        RED RETROSPOT SHOPPER BAG        10
…          …       …                                  …       …
1067366  581587     22899     CHILDREN'S APRON DOLLY GIRL          6
1067367  581587     23254    CHILDRENS CUTLERY DOLLY GIRL          4
1067368  581587     23255  CHILDRENS CUTLERY CIRCUS PARADE         4
1067369  581587     22138    BAKING SET 9 PIECE RETROSPOT          3
1067370  581587      POST                           POSTAGE         1

                 InvoiceDate  Price  Customer ID         Country  \
567942   2011-01-04 10:00:00   1.95      13313.0  United Kingdom
567943   2011-01-04 10:00:00   0.42      13313.0  United Kingdom
567944   2011-01-04 10:00:00   0.42      13313.0  United Kingdom
567945   2011-01-04 10:00:00   2.10      13313.0  United Kingdom
567946   2011-01-04 10:00:00   1.25      13313.0  United Kingdom
…                      …    …          …              …
1067366  2011-12-09 12:50:00   2.10      12680.0          France
1067367  2011-12-09 12:50:00   4.15      12680.0          France
1067368  2011-12-09 12:50:00   4.15      12680.0          France
1067369  2011-12-09 12:50:00   4.95      12680.0          France
1067370  2011-12-09 12:50:00  18.00      12680.0          France

                       date
567942  2011-01-04 10:00:00
567943  2011-01-04 10:00:00
567944  2011-01-04 10:00:00
567945  2011-01-04 10:00:00
567946  2011-01-04 10:00:00
…                      …
1067366 2011-12-09 12:50:00
1067367 2011-12-09 12:50:00
1067368 2011-12-09 12:50:00
1067369 2011-12-09 12:50:00
1067370 2011-12-09 12:50:00

[375251 rows x 9 columns]
```

[ ]:
```
retail[retail.date>='2011'].date.value_counts()
```

[ ]:
```
2011-11-14 15:27:00    543
2011-11-28 15:54:00    534
2011-12-05 17:17:00    530
2011-11-23 13:39:00    444
2011-10-31 14:09:00    436
                      …
2011-08-18 06:21:00      1
2011-06-20 16:05:00      1
2011-11-08 12:17:00      1
2011-05-03 12:51:00      1
2011-08-11 15:45:00      1
Name: date, Length: 18909, dtype: int64
```

0.3 Imputations

impute with filter

we can filter between August 1st and 30

[ ]:
```
retail[(retail.date >='2011-Aug-01') &(retail.date >='2011-Aug-30')]
```

[ ]:
```
         Invoice StockCode                          Description  Quantity  \
841626   C564748     23155  KNICKERBOCKERGLORY MAGNET ASSORTED         -6
841627   C564748     23211       RED ROCKING HORSE HAND PAINTED       -24
841628   C564748     23210     WHITE ROCKING HORSE HAND PAINTED       -24
841629   C564748     22978                    PANTRY ROLLING PIN        -5
841630   C564748     21672   WHITE SPOT RED CERAMIC DRAWER KNOB       -24
…          …       …                                      …       …
1067366   581587     22899         CHILDREN'S APRON DOLLY GIRL          6
1067367   581587     23254        CHILDRENS CUTLERY DOLLY GIRL          4
1067368   581587     23255      CHILDRENS CUTLERY CIRCUS PARADE         4
1067369   581587     22138        BAKING SET 9 PIECE RETROSPOT          3
1067370   581587      POST                              POSTAGE         1

                 InvoiceDate  Price  Customer ID Country                date
841626   2011-08-30 08:00:00   0.83      14911.0    EIRE 2011-08-30 08:00:00
841627   2011-08-30 08:00:00   1.25      14911.0    EIRE 2011-08-30 08:00:00
841628   2011-08-30 08:00:00   1.25      14911.0    EIRE 2011-08-30 08:00:00
841629   2011-08-30 08:00:00   3.75      14911.0    EIRE 2011-08-30 08:00:00
841630   2011-08-30 08:00:00   1.25      14911.0    EIRE 2011-08-30 08:00:00
…                      …    …           …      …                  …
1067366  2011-12-09 12:50:00   2.10      12680.0  France 2011-12-09 12:50:00
1067367  2011-12-09 12:50:00   4.15      12680.0  France 2011-12-09 12:50:00
1067368  2011-12-09 12:50:00   4.15      12680.0  France 2011-12-09 12:50:00
1067369  2011-12-09 12:50:00   4.95      12680.0  France 2011-12-09 12:50:00
1067370  2011-12-09 12:50:00  18.00      12680.0  France 2011-12-09 12:50:00
```

```
[173270 rows x 9 columns]
```

Impute or change a value

```
[ ]: # Get the column country, in this coulumn, when the value is EIRE make is =␣
     ↪Eastern Ireland
     retail['Country'][retail.Country=='EIRE']='Eastern Ireland'
```

```
[ ]: retail['StockCode'][retail.StockCode=='POST']='post'
```

```
[ ]: retail[retail.StockCode=='post']
```

```
[ ]:          Invoice StockCode Description  Quantity          InvoiceDate   Price  \
     89        489439      post     POSTAGE         3  2009-12-01 09:28:00   18.00
     126       489444      post     POSTAGE         1  2009-12-01 09:55:00  141.00
     173       489447      post     POSTAGE         1  2009-12-01 10:10:00  130.00
     625       489526      post     POSTAGE         6  2009-12-01 11:50:00   18.00
     927      C489538      post     POSTAGE        -1  2009-12-01 12:18:00    9.58
     …           …       …          …       …                 …     …
     1066677   581494      post     POSTAGE         2  2011-12-09 10:13:00   18.00
     1067191   581570      post     POSTAGE         1  2011-12-09 11:59:00   18.00
     1067228   581574      post     POSTAGE         2  2011-12-09 12:09:00   18.00
     1067229   581578      post     POSTAGE         3  2011-12-09 12:16:00   18.00
     1067370   581587      post     POSTAGE         1  2011-12-09 12:50:00   18.00

              Customer ID         Country                date
     89           12682.0          France 2009-12-01 09:28:00
     126          12636.0             USA 2009-12-01 09:55:00
     173          12362.0         Belgium 2009-12-01 10:10:00
     625          12533.0         Germany 2009-12-01 11:50:00
     927          15796.0  United Kingdom 2009-12-01 12:18:00
     …              …              …                   …
     1066677      12518.0         Germany 2011-12-09 10:13:00
     1067191      12662.0         Germany 2011-12-09 11:59:00
     1067228      12526.0         Germany 2011-12-09 12:09:00
     1067229      12713.0         Germany 2011-12-09 12:16:00
     1067370      12680.0          France 2011-12-09 12:50:00

     [1983 rows x 9 columns]
```

0.4 Indexing

This means having an address for your roles and columns. * We can have multiple indexes

```
[ ]: retail=retail.set_index('date')
```

This allows me to filter my data with index

[ ]:
```
retail['2011']
```

```
C:\Users\aduzo\Anaconda3\lib\site-packages\ipykernel_launcher.py:1:
FutureWarning: Indexing a DataFrame with a datetimelike index using a single
string to slice the rows, like `frame[string]`, is deprecated and will be
removed in a future version. Use `frame.loc[string]` instead.
  """Entry point for launching an IPython kernel.
```

[ ]:

date	Invoice	StockCode	Description	Quantity	InvoiceDate	Price	Customer ID	Country
2011-01-04 10:00:00	539993	22386	JUMBO BAG PINK POLKADOT	10	2011-01-04 10:00:00	1.95	13313.0	United Kingdom
2011-01-04 10:00:00	539993	21499	BLUE POLKADOT WRAP	25	2011-01-04 10:00:00	0.42	13313.0	United Kingdom
2011-01-04 10:00:00	539993	21498	RED RETROSPOT WRAP	25	2011-01-04 10:00:00	0.42	13313.0	United Kingdom
2011-01-04 10:00:00	539993	22379	RECYCLING BAG RETROSPOT	5	2011-01-04 10:00:00	2.10	13313.0	United Kingdom
2011-01-04 10:00:00	539993	20718	RED RETROSPOT SHOPPER BAG	10	2011-01-04 10:00:00	1.25	13313.0	United Kingdom
…	…	…	…	…	…	…	…	…
2011-12-09 12:50:00	581587	22899	CHILDREN'S APRON DOLLY GIRL	6	2011-12-09 12:50:00	2.10	12680.0	France
2011-12-09 12:50:00	581587	23254	CHILDRENS CUTLERY DOLLY GIRL	4	2011-12-09 12:50:00	4.15	12680.0	France
2011-12-09 12:50:00	581587	23255	CHILDRENS CUTLERY CIRCUS PARADE	4	2011-12-09 12:50:00	4.15	12680.0	France
2011-12-09 12:50:00	581587	22138	BAKING SET 9 PIECE RETROSPOT	3	2011-12-09 12:50:00	4.95	12680.0	France
2011-12-09 12:50:00	581587	post	POSTAGE	1	2011-12-09 12:50:00	18.00	12680.0	France

```
2011-12-09 12:50:00          France

[375251 rows x 8 columns]
```

This was posible because I have my data as an index, I am getting row index.

[ ]:
```
retail['2011-Jul-4']
```

```
C:\Users\aduzo\Anaconda3\lib\site-packages\ipykernel_launcher.py:1:
FutureWarning: Indexing a DataFrame with a datetimelike index using a single
string to slice the rows, like `frame[string]`, is deprecated and will be
removed in a future version. Use `frame.loc[string]` instead.
  """Entry point for launching an IPython kernel.
```

[ ]:
```
                     Invoice StockCode                          Description  \
date
2011-07-04 10:20:00  558775    48173C                DOORMAT BLACK FLOCK
2011-07-04 10:20:00  558775     48194                     DOORMAT HEARTS
2011-07-04 10:20:00  558775     48188             DOORMAT WELCOME PUPPIES
2011-07-04 10:20:00  558775     48187                 DOORMAT NEW ENGLAND
2011-07-04 10:20:00  558775     21523  DOORMAT FANCY FONT HOME SWEET HOME
…                     …       …                                      …
2011-07-04 16:35:00  558906     22382          LUNCH BAG SPACEBOY DESIGN
2011-07-04 16:35:00  558906     22383              LUNCH BAG SUKI DESIGN
2011-07-04 16:35:00  558906     21034        REX CASH+CARRY JUMBO SHOPPER
2011-07-04 16:35:00  558906     17003                 BROCADE RING PURSE
2011-07-04 16:35:00  558906     21933     PINK VINTAGE PAISLEY PICNIC BAG

                     Quantity          InvoiceDate  Price  Customer ID  \
date
2011-07-04 10:20:00       300  2011-07-04 10:20:00   4.58      18102.0
2011-07-04 10:20:00       300  2011-07-04 10:20:00   4.58      18102.0
2011-07-04 10:20:00       300  2011-07-04 10:20:00   4.58      18102.0
2011-07-04 10:20:00       300  2011-07-04 10:20:00   4.58      18102.0
2011-07-04 10:20:00       300  2011-07-04 10:20:00   4.58      18102.0
…                         …                    …    …          …
2011-07-04 16:35:00         6  2011-07-04 16:35:00   1.65      15555.0
2011-07-04 16:35:00         4  2011-07-04 16:35:00   1.65      15555.0
2011-07-04 16:35:00         4  2011-07-04 16:35:00   0.95      15555.0
2011-07-04 16:35:00        36  2011-07-04 16:35:00   0.29      15555.0
2011-07-04 16:35:00         3  2011-07-04 16:35:00   1.65      15555.0

                            Country
date
2011-07-04 10:20:00  United Kingdom
2011-07-04 10:20:00  United Kingdom
2011-07-04 10:20:00  United Kingdom
2011-07-04 10:20:00  United Kingdom
```

```
2011-07-04 10:20:00  United Kingdom
…                                …
2011-07-04 16:35:00  United Kingdom
2011-07-04 16:35:00  United Kingdom
2011-07-04 16:35:00  United Kingdom
2011-07-04 16:35:00  United Kingdom
2011-07-04 16:35:00  United Kingdom

[934 rows x 8 columns]
```

[ ]:
```
## remove index
retail=retail.reset_index()
```

Make 2 Index * County and date index with country being the outer index and date the lower index.

[ ]:
```
# Multi-level indexing
retail=retail.set_index(['Country','date'])
```

[ ]:
```
retail
```

[ ]:
```
                                    Invoice StockCode  \
Country        date
United Kingdom 2009-12-01 07:45:00  489434     85048
               2009-12-01 07:45:00  489434    79323P
               2009-12-01 07:45:00  489434    79323W
               2009-12-01 07:45:00  489434     22041
               2009-12-01 07:45:00  489434     21232
…                                     …       …
France         2011-12-09 12:50:00  581587     22899
               2011-12-09 12:50:00  581587     23254
               2011-12-09 12:50:00  581587     23255
               2011-12-09 12:50:00  581587     22138
               2011-12-09 12:50:00  581587      post

                                                                  Description  \
Country        date
United Kingdom 2009-12-01 07:45:00  15CM CHRISTMAS GLASS BALL 20 LIGHTS
               2009-12-01 07:45:00                   PINK CHERRY LIGHTS
               2009-12-01 07:45:00                  WHITE CHERRY LIGHTS
               2009-12-01 07:45:00        RECORD FRAME 7" SINGLE SIZE
               2009-12-01 07:45:00      STRAWBERRY CERAMIC TRINKET BOX
…                                                                    …
France         2011-12-09 12:50:00        CHILDREN'S APRON DOLLY GIRL
               2011-12-09 12:50:00       CHILDRENS CUTLERY DOLLY GIRL
               2011-12-09 12:50:00     CHILDRENS CUTLERY CIRCUS PARADE
               2011-12-09 12:50:00       BAKING SET 9 PIECE RETROSPOT
               2011-12-09 12:50:00                            POSTAGE
```

```
                                    Quantity          InvoiceDate  Price  \
Country        date
United Kingdom 2009-12-01 07:45:00        12  2009-12-01 07:45:00   6.95
               2009-12-01 07:45:00        12  2009-12-01 07:45:00   6.75
               2009-12-01 07:45:00        12  2009-12-01 07:45:00   6.75
               2009-12-01 07:45:00        48  2009-12-01 07:45:00   2.10
               2009-12-01 07:45:00        24  2009-12-01 07:45:00   1.25
…                                        …                   …    …
France         2011-12-09 12:50:00         6  2011-12-09 12:50:00   2.10
               2011-12-09 12:50:00         4  2011-12-09 12:50:00   4.15
               2011-12-09 12:50:00         4  2011-12-09 12:50:00   4.15
               2011-12-09 12:50:00         3  2011-12-09 12:50:00   4.95
               2011-12-09 12:50:00         1  2011-12-09 12:50:00  18.00

                                    Customer ID
Country        date
United Kingdom 2009-12-01 07:45:00      13085.0
               2009-12-01 07:45:00      13085.0
               2009-12-01 07:45:00      13085.0
               2009-12-01 07:45:00      13085.0
               2009-12-01 07:45:00      13085.0
…                                          …
France         2011-12-09 12:50:00      12680.0
               2011-12-09 12:50:00      12680.0
               2011-12-09 12:50:00      12680.0
               2011-12-09 12:50:00      12680.0
               2011-12-09 12:50:00      12680.0

[797885 rows x 7 columns]
```

0.5 Slicing Index

[ ]:
```
# Sort country and date
retail_sorted=retail.sort_index()
```

[ ]:
```
retail_sorted
```

[ ]:
```
                                Invoice StockCode  \
Country     date
Australia   2009-12-01 10:33:00  C489449     22087
            2009-12-01 10:33:00  C489449    85206A
            2009-12-01 10:33:00  C489449     21895
            2009-12-01 10:33:00  C489449     21896
            2009-12-01 10:33:00  C489449     22083
…                                  …       …
West Indies 2010-08-23 11:58:00   520018     20733
```

```
                    2010-08-23 11:58:00   520018     20734
                    2010-08-23 11:58:00   520018     20702
                    2010-08-23 11:58:00   520018     21678
                    2010-08-23 11:58:00   520018    84206A

                                                          Description  Quantity  \
Country     date
Australia   2009-12-01 10:33:00          PAPER BUNTING WHITE LACE       -12
            2009-12-01 10:33:00      CREAM FELT EASTER EGG BASKET        -6
            2009-12-01 10:33:00     POTTING SHED SOW 'N' GROW SET        -4
            2009-12-01 10:33:00                POTTING SHED TWINE        -6
            2009-12-01 10:33:00        PAPER CHAIN KIT RETRO SPOT       -12
…                                                                 …       …
West Indies 2010-08-23 11:58:00           GOLD MINI TAPE MEASURE          3
            2010-08-23 11:58:00         SILVER MINI TAPE MEASURE          3
            2010-08-23 11:58:00                PINK PADDED MOBILE         2
            2010-08-23 11:58:00         PAISLEY PATTERN  STICKERS         6
            2010-08-23 11:58:00  3 BLACK CATS W HEARTS BLANK CARD         6

                                          InvoiceDate  Price  Customer ID
Country     date
Australia   2009-12-01 10:33:00  2009-12-01 10:33:00   2.95      16321.0
            2009-12-01 10:33:00  2009-12-01 10:33:00   1.65      16321.0
            2009-12-01 10:33:00  2009-12-01 10:33:00   4.25      16321.0
            2009-12-01 10:33:00  2009-12-01 10:33:00   2.10      16321.0
            2009-12-01 10:33:00  2009-12-01 10:33:00   2.95      16321.0
…                                                  …    …          …
West Indies 2010-08-23 11:58:00  2010-08-23 11:58:00   0.85      18140.0
            2010-08-23 11:58:00  2010-08-23 11:58:00   0.85      18140.0
            2010-08-23 11:58:00  2010-08-23 11:58:00   4.25      18140.0
            2010-08-23 11:58:00  2010-08-23 11:58:00   0.85      18140.0
            2010-08-23 11:58:00  2010-08-23 11:58:00   0.19      18140.0

[797885 rows x 7 columns]
```

Now the row side has 2 levels country and date * Since we have already indexed this, this will help us.

[ ]:
```
# We can do filteration or slicing to get the items that we need.
retail_sorted.loc[('France','2011'),'Customer ID']
```

[ ]:
```
Country  date
France   2011-01-05 11:13:00    12494.0
         2011-01-05 11:36:00    12683.0
         2011-01-05 12:42:00    12681.0
         2011-01-05 12:42:00    12681.0
         2011-01-05 12:42:00    12681.0
                                  …
```

```
          2011-12-09 12:50:00    12680.0
          2011-12-09 12:50:00    12680.0
          2011-12-09 12:50:00    12680.0
          2011-12-09 12:50:00    12680.0
          2011-12-09 12:50:00    12680.0
Name: Customer ID, Length: 8037, dtype: float64
```

[ ]:
```
retail_sorted.loc[('France','2011'),['Customer ID','Description']]
```

[ ]:
```
                              Customer ID                         Description
Country date
France  2011-01-05 11:13:00      12494.0           RED RETROSPOT CAKE STAND
        2011-01-05 11:36:00      12683.0           RED RETROSPOT CAKE STAND
        2011-01-05 12:42:00      12681.0      GINGERBREAD MAN COOKIE CUTTER
        2011-01-05 12:42:00      12681.0                     CUTE CATS TAPE
        2011-01-05 12:42:00      12681.0        CABIN BAG VINTAGE RETROSPOT
…                                  …                                   …
        2011-12-09 12:50:00      12680.0       CHILDREN'S APRON DOLLY GIRL
        2011-12-09 12:50:00      12680.0      CHILDRENS CUTLERY DOLLY GIRL
        2011-12-09 12:50:00      12680.0    CHILDRENS CUTLERY CIRCUS PARADE
        2011-12-09 12:50:00      12680.0      BAKING SET 9 PIECE RETROSPOT
        2011-12-09 12:50:00      12680.0                           POSTAGE

[8037 rows x 2 columns]
```

Note:

Some number are not numbers, so you can have, {56,000,70,000} these commas can not be parsed inside python.

- replace , with nothing
- df[‘sales’]=df[‘sales’].str.replace(‘,’,”)
- df[’sales]= pd.to_numeric(df[‘sales’])

[ ]:
```
####group by and aggregations

#count() - Number of non-null observations
#sum() - Sum of values
#mean() - Mean of values
#median() - Arithmetic median of values
#min() - Minimum
#max() - Maximum
#mode() - Mode
#std() - Standard deviation
#var() - Variance
```

0.6 Groupby and Aggregations

[ ]:
```
retail= retail.reset_index()
```

[ ]:
```
retail.head()
```

[ ]:
```
          Country                date Invoice StockCode  \
0  United Kingdom 2009-12-01 07:45:00  489434     85048
1  United Kingdom 2009-12-01 07:45:00  489434    79323P
2  United Kingdom 2009-12-01 07:45:00  489434    79323W
3  United Kingdom 2009-12-01 07:45:00  489434     22041
4  United Kingdom 2009-12-01 07:45:00  489434     21232

                          Description  Quantity          InvoiceDate  Price  \
0  15CM CHRISTMAS GLASS BALL 20 LIGHTS        12  2009-12-01 07:45:00   6.95
1                   PINK CHERRY LIGHTS        12  2009-12-01 07:45:00   6.75
2                  WHITE CHERRY LIGHTS        12  2009-12-01 07:45:00   6.75
3         RECORD FRAME 7" SINGLE SIZE         48  2009-12-01 07:45:00   2.10
4       STRAWBERRY CERAMIC TRINKET BOX        24  2009-12-01 07:45:00   1.25

   Customer ID
0      13085.0
1      13085.0
2      13085.0
3      13085.0
4      13085.0
```

[ ]:
```
retail.groupby('Country').head()
```

[ ]:
```
                   Country                date Invoice StockCode  \
0           United Kingdom 2009-12-01 07:45:00  489434     85048
1           United Kingdom 2009-12-01 07:45:00  489434    79323P
2           United Kingdom 2009-12-01 07:45:00  489434    79323W
3           United Kingdom 2009-12-01 07:45:00  489434     22041
4           United Kingdom 2009-12-01 07:45:00  489434     21232
…                      …                  …     …        …
511759  European Community 2011-04-26 10:54:00  551013     22839
511760  European Community 2011-04-26 10:54:00  551013     22840
511761  European Community 2011-04-26 10:54:00  551013     22841
511762  European Community 2011-04-26 10:54:00  551013     22457
511763  European Community 2011-04-26 10:54:00  551013     22314

                                Description  Quantity          InvoiceDate  \
0       15CM CHRISTMAS GLASS BALL 20 LIGHTS        12  2009-12-01 07:45:00
1                     PINK CHERRY LIGHTS           12  2009-12-01 07:45:00
2                    WHITE CHERRY LIGHTS           12  2009-12-01 07:45:00
3              RECORD FRAME 7" SINGLE SIZE         48  2009-12-01 07:45:00
4            STRAWBERRY CERAMIC TRINKET BOX        24  2009-12-01 07:45:00
```

```
…                                        …       …                   …
511759      3 TIER CAKE TIN GREEN AND CREAM         1  2011-04-26 10:54:00
511760           ROUND CAKE TIN VINTAGE RED         2  2011-04-26 10:54:00
511761         ROUND CAKE TIN VINTAGE GREEN         2  2011-04-26 10:54:00
511762      NATURAL SLATE HEART CHALKBOARD          6  2011-04-26 10:54:00
511763          OFFICE MUG WARMER CHOC+BLUE         6  2011-04-26 10:54:00

        Price  Customer ID
0        6.95      13085.0
1        6.75      13085.0
2        6.75      13085.0
3        2.10      13085.0
4        1.25      13085.0
…        …         …
511759  14.95      15108.0
511760   7.95      15108.0
511761   7.95      15108.0
511762   2.95      15108.0
511763   2.95      15108.0

[205 rows x 9 columns]
```

[ ]:
```
# group by country and get total amount of sales for each country
## Total sales for each country
retail.groupby('Country').Quantity.sum()
```

[ ]:
```
Country
Australia               103375
Austria                  11306
Bahrain                    755
Belgium                  34598
Brazil                     545
Canada                    3657
Channel Islands          20387
Cyprus                   10652
Czech Republic             592
Denmark                  234764
Eastern Ireland          309717
European Community          497
Finland                  14317
France                  179959
Germany                 221816
Greece                    7707
Iceland                   2967
Israel                    5119
Italy                    15122
Japan                    30138
```

```
Korea                     598
Lebanon                   386
Lithuania                2306
Malta                    2491
Netherlands            381853
Nigeria                   103
Norway                  23528
Poland                   5504
Portugal                27072
RSA                       943
Saudi Arabia               75
Singapore                6987
Spain                   44667
Sweden                  87720
Switzerland             51721
Thailand                 2552
USA                      3700
United Arab Emirates     5721
United Kingdom        8194318
Unspecified              5099
West Indies               395
Name: Quantity, dtype: int64
```

[ ]:
```
## OR
retail.groupby('Country')['Quantity'].sum()
```

[ ]:
```
Country
Australia               103375
Austria                  11306
Bahrain                    755
Belgium                  34598
Brazil                     545
Canada                    3657
Channel Islands          20387
Cyprus                   10652
Czech Republic             592
Denmark                 234764
Eastern Ireland         309717
European Community         497
Finland                  14317
France                  179959
Germany                 221816
Greece                    7707
Iceland                   2967
Israel                    5119
Italy                    15122
Japan                    30138
```

```
Korea                     598
Lebanon                   386
Lithuania                2306
Malta                    2491
Netherlands            381853
Nigeria                   103
Norway                  23528
Poland                   5504
Portugal                27072
RSA                       943
Saudi Arabia               75
Singapore                6987
Spain                   44667
Sweden                  87720
Switzerland             51721
Thailand                 2552
USA                      3700
United Arab Emirates     5721
United Kingdom        8194318
Unspecified              5099
West Indies               395
Name: Quantity, dtype: int64
```

[ ]:
```
retail.groupby(['Country','Description'])['Quantity','Price'].mean()
```

```
C:\Users\aduzo\Anaconda3\lib\site-packages\ipykernel_launcher.py:1:
FutureWarning: Indexing with multiple keys (implicitly converted to a tuple of
keys) will be deprecated, use a list instead.
  """Entry point for launching an IPython kernel.
```

[ ]:

Country	Description	Quantity	Price
Australia	DOLLY GIRL BEAKER	200.0	1.08
	I LOVE LONDON MINI BACKPACK	4.0	4.15
	10 COLOUR SPACEBOY PEN	48.0	0.85
	12 PENCIL SMALL TUBE WOODLAND	384.0	0.55
	12 PENCILS SMALL TUBE RED SPOTTY	24.0	0.65
…		…	…
West Indies	VINTAGE BEAD PINK SCARF	3.0	7.95
	WHITE AND BLUE CERAMIC OIL BURNER	6.0	1.25
	WOODLAND PARTY BAG + STICKER SET	1.0	1.65
	WOVEN BERRIES CUSHION COVER	2.0	4.95
	WOVEN FROST CUSHION COVER	2.0	4.95

[29514 rows x 2 columns]

0.6.1 Slicing the group by

making multiple aggregations, using aggregation function

[ ]:
```
## making 2 operations
a=retail.groupby(['Country','Description'])['Quantity','Price'].agg([np.mean,np.
 ↪median])
a
```

```
C:\Users\aduzo\Anaconda3\lib\site-packages\ipykernel_launcher.py:2:
FutureWarning: Indexing with multiple keys (implicitly converted to a tuple of
keys) will be deprecated, use a list instead.
```

[ ]:

		Quantity		Price	
		mean	median	mean	median
Country	Description				
Australia	DOLLY GIRL BEAKER	200.0	200.0	1.08	1.08
	I LOVE LONDON MINI BACKPACK	4.0	4.0	4.15	4.15
	10 COLOUR SPACEBOY PEN	48.0	48.0	0.85	0.85
	12 PENCIL SMALL TUBE WOODLAND	384.0	384.0	0.55	0.55
	12 PENCILS SMALL TUBE RED SPOTTY	24.0	24.0	0.65	0.65
…		…	…	…	…
West Indies	VINTAGE BEAD PINK SCARF	3.0	3.0	7.95	7.95
	WHITE AND BLUE CERAMIC OIL BURNER	6.0	6.0	1.25	1.25
	WOODLAND PARTY BAG + STICKER SET	1.0	1.0	1.65	1.65
	WOVEN BERRIES CUSHION COVER	2.0	2.0	4.95	4.95
	WOVEN FROST CUSHION COVER	2.0	2.0	4.95	4.95

[29514 rows x 4 columns]

[ ]:
```
## Slicing
a=a.sort_index()
# row side, column side a.loc[(row_index,row_value),(col_index,col_name)]
a.loc[('Australia','10 COLOUR SPACEBOY PEN'),('Price','median')]
```

[ ]: 0.85

0.7 Droping Levels

[ ]:
```
## Without index, change to normal table
b=retail.groupby(['Country','Description'])['Quantity','Price'].agg([np.mean,np.
 ↪median]).reset_index()
b
```

```
C:\Users\aduzo\Anaconda3\lib\site-packages\ipykernel_launcher.py:2:
FutureWarning: Indexing with multiple keys (implicitly converted to a tuple of
keys) will be deprecated, use a list instead.
```

[ ]:

	Country	Description	Quantity		Price	
			mean	median	mean	
0	Australia	DOLLY GIRL BEAKER	200.0	200.0	1.08	1.08
1	Australia	I LOVE LONDON MINI BACKPACK	4.0	4.0	4.15	4.15
2	Australia	10 COLOUR SPACEBOY PEN	48.0	48.0	0.85	0.85
3	Australia	12 PENCIL SMALL TUBE WOODLAND	384.0	384.0	0.55	0.55
4	Australia	12 PENCILS SMALL TUBE RED SPOTTY	24.0	24.0	0.65	0.65
…	…	…	…	…	…	…
29509	West Indies	VINTAGE BEAD PINK SCARF	3.0	3.0	7.95	7.95
29510	West Indies	WHITE AND BLUE CERAMIC OIL BURNER	6.0	6.0	1.25	1.25
29511	West Indies	WOODLAND PARTY BAG + STICKER SET	1.0	1.0	1.65	1.65
29512	West Indies	WOVEN BERRIES CUSHION COVER	2.0	2.0	4.95	4.95
29513	West Indies	WOVEN FROST CUSHION COVER	2.0	2.0	4.95	4.95

	median
0	1.08
1	4.15
2	0.85
3	0.55
4	0.65
…	…
29509	7.95
29510	1.25
29511	1.65
29512	4.95
29513	4.95

[29514 rows x 6 columns]

[ ]:
```
# drop column index at level 0
b=b.droplevel(axis=1,level=0)
```

[ ]:
```
b
```

[ ]:

			mean	median	mean	
0	Australia	DOLLY GIRL BEAKER	200.0	200.0	1.08	1.08
1	Australia	I LOVE LONDON MINI BACKPACK	4.0	4.0	4.15	4.15
2	Australia	10 COLOUR SPACEBOY PEN	48.0	48.0	0.85	0.85
3	Australia	12 PENCIL SMALL TUBE WOODLAND	384.0	384.0	0.55	0.55
4	Australia	12 PENCILS SMALL TUBE RED SPOTTY	24.0	24.0	0.65	0.65
…	…	…	…	…	…	…
29509	West Indies	VINTAGE BEAD PINK SCARF	3.0	3.0	7.95	7.95
29510	West Indies	WHITE AND BLUE CERAMIC OIL BURNER	6.0	6.0	1.25	1.25
29511	West Indies	WOODLAND PARTY BAG + STICKER SET	1.0	1.0	1.65	1.65
29512	West Indies	WOVEN BERRIES CUSHION COVER	2.0	2.0	4.95	4.95
29513	West Indies	WOVEN FROST CUSHION COVER	2.0	2.0	4.95	4.95

```
       median
0        1.08
1        4.15
2        0.85
3        0.55
4        0.65
…         …
29509    7.95
29510    1.25
29511    1.65
29512    4.95
29513    4.95

[29514 rows x 6 columns]
```

0.8 The proper form

The identity of the columns where lost

[ ]:
```
retail.groupby(['Country','Description']).agg(mean_qty=('Quantity',np.mean),
                                               mean_price=('Price',np.mean),
                                               median_qty=('Quantity',np.median),
                                               median_price=('Price',np.median))
```

[ ]:
```
                                                mean_qty  mean_price  \
Country     Description
Australia    DOLLY GIRL BEAKER                     200.0        1.08
             I LOVE LONDON MINI BACKPACK             4.0        4.15
            10 COLOUR SPACEBOY PEN                  48.0        0.85
            12 PENCIL SMALL TUBE WOODLAND          384.0        0.55
            12 PENCILS SMALL TUBE RED SPOTTY        24.0        0.65
…                                                   …           …
West Indies VINTAGE BEAD PINK SCARF                  3.0        7.95
            WHITE AND BLUE CERAMIC OIL BURNER        6.0        1.25
            WOODLAND PARTY BAG + STICKER SET         1.0        1.65
            WOVEN BERRIES CUSHION COVER              2.0        4.95
            WOVEN FROST CUSHION COVER                2.0        4.95

                                                median_qty  median_price
Country     Description
Australia    DOLLY GIRL BEAKER                       200.0          1.08
             I LOVE LONDON MINI BACKPACK               4.0          4.15
            10 COLOUR SPACEBOY PEN                    48.0          0.85
            12 PENCIL SMALL TUBE WOODLAND            384.0          0.55
            12 PENCILS SMALL TUBE RED SPOTTY          24.0          0.65
…                                                     …             …
```

```
West Indies VINTAGE BEAD PINK SCARF                     3.0          7.95
            WHITE AND BLUE CERAMIC OIL BURNER           6.0          1.25
            WOODLAND PARTY BAG + STICKER SET            1.0          1.65
            WOVEN BERRIES CUSHION COVER                 2.0          4.95
            WOVEN FROST CUSHION COVER                   2.0          4.95

[29514 rows x 4 columns]
```

[ ]:
```
## remove indexes
retail.groupby(['Country','Description']).agg(mean_qty=('Quantity',np.mean),
                                               mean_price=('Price',np.mean),
                                               median_qty=('Quantity',np.median),
                                               median_price=('Price',np.median)).
 ↪reset_index()
```

[ ]:
```
           Country                         Description  mean_qty  mean_price  \
0        Australia                   DOLLY GIRL BEAKER     200.0        1.08
1        Australia          I LOVE LONDON MINI BACKPACK       4.0        4.15
2        Australia               10 COLOUR SPACEBOY PEN      48.0        0.85
3        Australia        12 PENCIL SMALL TUBE WOODLAND     384.0        0.55
4        Australia     12 PENCILS SMALL TUBE RED SPOTTY      24.0        0.65
…              …                                    …         …           …
29509  West Indies             VINTAGE BEAD PINK SCARF       3.0        7.95
29510  West Indies    WHITE AND BLUE CERAMIC OIL BURNER      6.0        1.25
29511  West Indies     WOODLAND PARTY BAG + STICKER SET      1.0        1.65
29512  West Indies         WOVEN BERRIES CUSHION COVER       2.0        4.95
29513  West Indies            WOVEN FROST CUSHION COVER      2.0        4.95

       median_qty  median_price
0           200.0          1.08
1             4.0          4.15
2            48.0          0.85
3           384.0          0.55
4            24.0          0.65
…               …             …
29509         3.0          7.95
29510         6.0          1.25
29511         1.0          1.65
29512         2.0          4.95
29513         2.0          4.95

[29514 rows x 6 columns]
```

0.9 Pivot Tables

[ ]:
```
country_date_qty= retail[['Country','date','Quantity','Price']]
country_date_qty.head()
```

[ ]:

	Country	date	Quantity	Price
0	United Kingdom	2009-12-01 07:45:00	12	6.95
1	United Kingdom	2009-12-01 07:45:00	12	6.75
2	United Kingdom	2009-12-01 07:45:00	12	6.75
3	United Kingdom	2009-12-01 07:45:00	48	2.10
4	United Kingdom	2009-12-01 07:45:00	24	1.25

[ ]:
```
help(pd.pivot_table)
```

```
Help on function pivot_table in module pandas.core.reshape.pivot:

pivot_table(data, values=None, index=None, columns=None, aggfunc='mean',
fill_value=None, margins=False, dropna=True, margins_name='All', observed=False)
-> 'DataFrame'
    Create a spreadsheet-style pivot table as a DataFrame.

    The levels in the pivot table will be stored in MultiIndex objects
    (hierarchical indexes) on the index and columns of the result DataFrame.

    Parameters
    ----------
    data : DataFrame
    values : column to aggregate, optional
    index : column, Grouper, array, or list of the previous
        If an array is passed, it must be the same length as the data. The
        list can contain any of the other types (except list).
        Keys to group by on the pivot table index.  If an array is passed,
        it is being used as the same manner as column values.
    columns : column, Grouper, array, or list of the previous
        If an array is passed, it must be the same length as the data. The
        list can contain any of the other types (except list).
        Keys to group by on the pivot table column.  If an array is passed,
        it is being used as the same manner as column values.
    aggfunc : function, list of functions, dict, default numpy.mean
        If list of functions passed, the resulting pivot table will have
        hierarchical columns whose top level are the function names
        (inferred from the function objects themselves)
        If dict is passed, the key is column to aggregate and value
        is function or list of functions.
    fill_value : scalar, default None
        Value to replace missing values with (in the resulting pivot table,
        after aggregation).
    margins : bool, default False
```

```
    Add all row / columns (e.g. for subtotal / grand totals).
dropna : bool, default True
    Do not include columns whose entries are all NaN.
margins_name : str, default 'All'
    Name of the row / column that will contain the totals
    when margins is True.
observed : bool, default False
    This only applies if any of the groupers are Categoricals.
    If True: only show observed values for categorical groupers.
    If False: show all values for categorical groupers.

    .. versionchanged:: 0.25.0

Returns
-------
DataFrame
    An Excel style pivot table.

See Also
--------
DataFrame.pivot : Pivot without aggregation that can handle
    non-numeric data.
DataFrame.melt: Unpivot a DataFrame from wide to long format,
    optionally leaving identifiers set.
wide_to_long : Wide panel to long format. Less flexible but more
    user-friendly than melt.

Examples
--------
>>> df = pd.DataFrame({"A": ["foo", "foo", "foo", "foo", "foo",
…                          "bar", "bar", "bar", "bar"],
…                    "B": ["one", "one", "one", "two", "two",
…                          "one", "one", "two", "two"],
…                    "C": ["small", "large", "large", "small",
…                          "small", "large", "small", "small",
…                          "large"],
…                    "D": [1, 2, 2, 3, 3, 4, 5, 6, 7],
…                    "E": [2, 4, 5, 5, 6, 6, 8, 9, 9]})
>>> df
     A    B      C  D  E
0  foo  one  small  1  2
1  foo  one  large  2  4
2  foo  one  large  2  5
3  foo  two  small  3  5
4  foo  two  small  3  6
5  bar  one  large  4  6
6  bar  one  small  5  8
7  bar  two  small  6  9
```

```
8  bar  two  large  7  9

This first example aggregates values by taking the sum.

>>> table = pd.pivot_table(df, values='D', index=['A', 'B'],
…                     columns=['C'], aggfunc=np.sum)
>>> table
C        large  small
A   B
bar one    4.0    5.0
    two    7.0    6.0
foo one    4.0    1.0
    two    NaN    6.0

We can also fill missing values using the `fill_value` parameter.

>>> table = pd.pivot_table(df, values='D', index=['A', 'B'],
…                     columns=['C'], aggfunc=np.sum, fill_value=0)
>>> table
C        large  small
A   B
bar one      4      5
    two      7      6
foo one      4      1
    two      0      6

The next example aggregates by taking the mean across multiple columns.

>>> table = pd.pivot_table(df, values=['D', 'E'], index=['A', 'C'],
…                     aggfunc={'D': np.mean,
…                              'E': np.mean})
>>> table
                D         E
A   C
bar large  5.500000  7.500000
    small  5.500000  8.500000
foo large  2.000000  4.500000
    small  2.333333  4.333333

We can also calculate multiple types of aggregations for any given
value column.

>>> table = pd.pivot_table(df, values=['D', 'E'], index=['A', 'C'],
…                     aggfunc={'D': np.mean,
…                              'E': [min, max, np.mean]})
>>> table
                D    E
            mean  max      mean  min
```

```
        A   C
bar large  5.500000  9.0  7.500000  6.0
    small  5.500000  9.0  8.500000  8.0
foo large  2.000000  5.0  4.500000  4.0
    small  2.333333  6.0  4.333333  2.0
```

[ ]:
```
## I want to keep the country in columns
country_pivoted=pd.
 ↪pivot_table(country_date_qty,index='date',columns='Country',values='Quantity')
```

[ ]:
```
country_pivoted
```

[ ]:
```
Country              Australia  Austria  Bahrain  Belgium  Brazil  Canada  \
date
2009-12-01 07:45:00        NaN      NaN      NaN      NaN     NaN     NaN
2009-12-01 07:46:00        NaN      NaN      NaN      NaN     NaN     NaN
2009-12-01 09:06:00        NaN      NaN      NaN      NaN     NaN     NaN
2009-12-01 09:08:00        NaN      NaN      NaN      NaN     NaN     NaN
2009-12-01 09:24:00        NaN      NaN      NaN      NaN     NaN     NaN
…                          …        …        …        …       …       …
2011-12-09 12:23:00        NaN      NaN      NaN      NaN     NaN     NaN
2011-12-09 12:25:00        NaN      NaN      NaN      NaN     NaN     NaN
2011-12-09 12:31:00        NaN      NaN      NaN      NaN     NaN     NaN
2011-12-09 12:49:00        NaN      NaN      NaN      NaN     NaN     NaN
2011-12-09 12:50:00        NaN      NaN      NaN      NaN     NaN     NaN

Country              Channel Islands  Cyprus  Czech Republic  Denmark  …  \
date                                                                     …
2009-12-01 07:45:00              NaN     NaN             NaN      NaN  …
2009-12-01 07:46:00              NaN     NaN             NaN      NaN  …
2009-12-01 09:06:00              NaN     NaN             NaN      NaN  …
2009-12-01 09:08:00              NaN     NaN             NaN      NaN  …
2009-12-01 09:24:00              NaN     NaN             NaN      NaN  …
…                                …       …               …        …  …
2011-12-09 12:23:00              NaN     NaN             NaN      NaN  …
2011-12-09 12:25:00              NaN     NaN             NaN      NaN  …
2011-12-09 12:31:00              NaN     NaN             NaN      NaN  …
2011-12-09 12:49:00              NaN     NaN             NaN      NaN  …
2011-12-09 12:50:00              NaN     NaN             NaN      NaN  …

Country              Singapore  Spain  Sweden  Switzerland  Thailand  USA  \
date
2009-12-01 07:45:00        NaN    NaN     NaN          NaN       NaN  NaN
2009-12-01 07:46:00        NaN    NaN     NaN          NaN       NaN  NaN
2009-12-01 09:06:00        NaN    NaN     NaN          NaN       NaN  NaN
2009-12-01 09:08:00        NaN    NaN     NaN          NaN       NaN  NaN
```

```
2009-12-01 09:24:00        NaN    NaN     NaN           NaN       NaN  NaN
…                          …      …       …             …         …    …
2011-12-09 12:23:00        NaN    NaN     NaN           NaN       NaN  NaN
2011-12-09 12:25:00        NaN    NaN     NaN           NaN       NaN  NaN
2011-12-09 12:31:00        NaN    NaN     NaN           NaN       NaN  NaN
2011-12-09 12:49:00        NaN    NaN     NaN           NaN       NaN  NaN
2011-12-09 12:50:00        NaN    NaN     NaN           NaN       NaN  NaN

Country              United Arab Emirates  United Kingdom  Unspecified  \
date
2009-12-01 07:45:00                   NaN       20.750000          NaN
2009-12-01 07:46:00                   NaN       15.000000          NaN
2009-12-01 09:06:00                   NaN       10.157895          NaN
2009-12-01 09:08:00                   NaN        6.304348          NaN
2009-12-01 09:24:00                   NaN       48.588235          NaN
…                                     …              …           …
2011-12-09 12:23:00                   NaN       38.000000          NaN
2011-12-09 12:25:00                   NaN       60.000000          NaN
2011-12-09 12:31:00                   NaN       13.238095          NaN
2011-12-09 12:49:00                   NaN       16.500000          NaN
2011-12-09 12:50:00                   NaN             NaN          NaN

Country              West Indies
date
2009-12-01 07:45:00          NaN
2009-12-01 07:46:00          NaN
2009-12-01 09:06:00          NaN
2009-12-01 09:08:00          NaN
2009-12-01 09:24:00          NaN
…                            …
2011-12-09 12:23:00          NaN
2011-12-09 12:25:00          NaN
2011-12-09 12:31:00          NaN
2011-12-09 12:49:00          NaN
2011-12-09 12:50:00          NaN

[41439 rows x 41 columns]
```

Alot of NANs, means at this time, there was no sales

[ ]:
```
# If you don't want NANs you do fill_value=0

country_pivoted=pd.
 ↪pivot_table(country_date_qty,index='date',columns='Country',values='Quantity',fill_value=0)
```

[ ]:
```
country_pivoted
```

```
[ ]: Country              Australia  Austria  Bahrain  Belgium  Brazil  Canada  \
     date
     2009-12-01 07:45:00        0.0      0.0      0.0      0.0     0.0     0.0
     2009-12-01 07:46:00        0.0      0.0      0.0      0.0     0.0     0.0
     2009-12-01 09:06:00        0.0      0.0      0.0      0.0     0.0     0.0
     2009-12-01 09:08:00        0.0      0.0      0.0      0.0     0.0     0.0
     2009-12-01 09:24:00        0.0      0.0      0.0      0.0     0.0     0.0
     …                          …        …        …        …       …       …
     2011-12-09 12:23:00        0.0      0.0      0.0      0.0     0.0     0.0
     2011-12-09 12:25:00        0.0      0.0      0.0      0.0     0.0     0.0
     2011-12-09 12:31:00        0.0      0.0      0.0      0.0     0.0     0.0
     2011-12-09 12:49:00        0.0      0.0      0.0      0.0     0.0     0.0
     2011-12-09 12:50:00        0.0      0.0      0.0      0.0     0.0     0.0

     Country              Channel Islands  Cyprus  Czech Republic  Denmark  …  \
     date                                                                  …
     2009-12-01 07:45:00              0.0     0.0             0.0      0.0  …
     2009-12-01 07:46:00              0.0     0.0             0.0      0.0  …
     2009-12-01 09:06:00              0.0     0.0             0.0      0.0  …
     2009-12-01 09:08:00              0.0     0.0             0.0      0.0  …
     2009-12-01 09:24:00              0.0     0.0             0.0      0.0  …
     …                                …       …               …        …  …
     2011-12-09 12:23:00              0.0     0.0             0.0      0.0  …
     2011-12-09 12:25:00              0.0     0.0             0.0      0.0  …
     2011-12-09 12:31:00              0.0     0.0             0.0      0.0  …
     2011-12-09 12:49:00              0.0     0.0             0.0      0.0  …
     2011-12-09 12:50:00              0.0     0.0             0.0      0.0  …

     Country              Singapore  Spain  Sweden  Switzerland  Thailand  USA  \
     date
     2009-12-01 07:45:00        0.0    0.0     0.0          0.0       0.0  0.0
     2009-12-01 07:46:00        0.0    0.0     0.0          0.0       0.0  0.0
     2009-12-01 09:06:00        0.0    0.0     0.0          0.0       0.0  0.0
     2009-12-01 09:08:00        0.0    0.0     0.0          0.0       0.0  0.0
     2009-12-01 09:24:00        0.0    0.0     0.0          0.0       0.0  0.0
     …                          …      …       …            …         …    …
     2011-12-09 12:23:00        0.0    0.0     0.0          0.0       0.0  0.0
     2011-12-09 12:25:00        0.0    0.0     0.0          0.0       0.0  0.0
     2011-12-09 12:31:00        0.0    0.0     0.0          0.0       0.0  0.0
     2011-12-09 12:49:00        0.0    0.0     0.0          0.0       0.0  0.0
     2011-12-09 12:50:00        0.0    0.0     0.0          0.0       0.0  0.0

     Country              United Arab Emirates  United Kingdom  Unspecified  \
     date
     2009-12-01 07:45:00                   0.0       20.750000          0.0
     2009-12-01 07:46:00                   0.0       15.000000          0.0
     2009-12-01 09:06:00                   0.0       10.157895          0.0
```

```
2009-12-01 09:08:00                  0.0        6.304348          0.0
2009-12-01 09:24:00                  0.0       48.588235          0.0
…                                    …              …            …
2011-12-09 12:23:00                  0.0       38.000000          0.0
2011-12-09 12:25:00                  0.0       60.000000          0.0
2011-12-09 12:31:00                  0.0       13.238095          0.0
2011-12-09 12:49:00                  0.0       16.500000          0.0
2011-12-09 12:50:00                  0.0        0.000000          0.0

Country              West Indies
date
2009-12-01 07:45:00          0.0
2009-12-01 07:46:00          0.0
2009-12-01 09:06:00          0.0
2009-12-01 09:08:00          0.0
2009-12-01 09:24:00          0.0
…                            …
2011-12-09 12:23:00          0.0
2011-12-09 12:25:00          0.0
2011-12-09 12:31:00          0.0
2011-12-09 12:49:00          0.0
2011-12-09 12:50:00          0.0

[41439 rows x 41 columns]
```

- Index has not been reset
- Also add aggfun=np.sum to sum the Quantity values by date

in country_pivoted, you can see that the sales are continues if we do not add aggfun, it does not look right.This is because we need to tell python that we want to sum the quantity.

[ ]:
```
country_pivoted= pd.pivot_table(country_date_qty,index='date',columns='Country',
                                values='Quantity',fill_value=0,aggfunc=np.sum).
↪reset_index()
```

Make timeseries only for UK

[ ]:
```
country_pivoted
```

[ ]:
```
Country                date  Australia  Austria  Bahrain  Belgium  Brazil  \
0       2009-12-01 07:45:00          0        0        0        0       0
1       2009-12-01 07:46:00          0        0        0        0       0
2       2009-12-01 09:06:00          0        0        0        0       0
3       2009-12-01 09:08:00          0        0        0        0       0
4       2009-12-01 09:24:00          0        0        0        0       0
…                       …        …        …        …        …       …
41434   2011-12-09 12:23:00          0        0        0        0       0
41435   2011-12-09 12:25:00          0        0        0        0       0
41436   2011-12-09 12:31:00          0        0        0        0       0
```

```
41437   2011-12-09 12:49:00          0        0        0        0       0
41438   2011-12-09 12:50:00          0        0        0        0       0

Country  Canada  Channel Islands  Cyprus  Czech Republic  …  Singapore  \
0             0                0       0               0  …          0
1             0                0       0               0  …          0
2             0                0       0               0  …          0
3             0                0       0               0  …          0
4             0                0       0               0  …          0
…           …                …       …               …  …        …
41434         0                0       0               0  …          0
41435         0                0       0               0  …          0
41436         0                0       0               0  …          0
41437         0                0       0               0  …          0
41438         0                0       0               0  …          0

Country  Spain  Sweden  Switzerland  Thailand  USA  United Arab Emirates  \
0            0       0            0         0    0                     0
1            0       0            0         0    0                     0
2            0       0            0         0    0                     0
3            0       0            0         0    0                     0
4            0       0            0         0    0                     0
…          …     …          …       …  …                   …
41434        0       0            0         0    0                     0
41435        0       0            0         0    0                     0
41436        0       0            0         0    0                     0
41437        0       0            0         0    0                     0
41438        0       0            0         0    0                     0

Country  United Kingdom  Unspecified  West Indies
0                   166            0            0
1                    60            0            0
2                   193            0            0
3                   145            0            0
4                   826            0            0
…                 …          …          …
41434                76            0            0
41435               120            0            0
41436               278            0            0
41437                66            0            0
41438                 0            0            0

[41439 rows x 42 columns]
```

[ ]:
```
country_pivoted[['date','United Kingdom']]
```

[ ]:
```
Country                date  United Kingdom
0       2009-12-01 07:45:00             166
1       2009-12-01 07:46:00              60
2       2009-12-01 09:06:00             193
3       2009-12-01 09:08:00             145
4       2009-12-01 09:24:00             826
…                       …               …
41434   2011-12-09 12:23:00              76
41435   2011-12-09 12:25:00             120
41436   2011-12-09 12:31:00             278
41437   2011-12-09 12:49:00              66
41438   2011-12-09 12:50:00               0

[41439 rows x 2 columns]
```

As seen above, column United Kingdom is not continues

0.10 Aggregate Function in Pivot table

[ ]:
```
country_date_qty=retail[['Country','date','Quantity','Price']]
country_date_qty
```

[ ]:
```
               Country                 date  Quantity  Price
0       United Kingdom 2009-12-01 07:45:00        12   6.95
1       United Kingdom 2009-12-01 07:45:00        12   6.75
2       United Kingdom 2009-12-01 07:45:00        12   6.75
3       United Kingdom 2009-12-01 07:45:00        48   2.10
4       United Kingdom 2009-12-01 07:45:00        24   1.25
…                  …                   …       …     …
797880          France 2011-12-09 12:50:00         6   2.10
797881          France 2011-12-09 12:50:00         4   4.15
797882          France 2011-12-09 12:50:00         4   4.15
797883          France 2011-12-09 12:50:00         3   4.95
797884          France 2011-12-09 12:50:00         1  18.00

[797885 rows x 4 columns]
```

[ ]:
```
country_pivoted= pd.pivot_table(country_date_qty,index='date',columns='Country',
↪values=['Quantity','Price'],fill_value=0,aggfunc=np.sum).reset_index()
```

[ ]:
```
country_pivoted
```

[ ]:
```
                      date     Price                                          \
Country                         Australia Austria Bahrain Belgium Brazil Canada
0       2009-12-01 07:45:00         0.0     0.0     0.0     0.0    0.0    0.0
1       2009-12-01 07:46:00         0.0     0.0     0.0     0.0    0.0    0.0
2       2009-12-01 09:06:00         0.0     0.0     0.0     0.0    0.0    0.0
```

```
3       2009-12-01 09:08:00       0.0     0.0     0.0     0.0    0.0    0.0
4       2009-12-01 09:24:00       0.0     0.0     0.0     0.0    0.0    0.0
…                       …       …     …     …     …    …    …
41434   2011-12-09 12:23:00       0.0     0.0     0.0     0.0    0.0    0.0
41435   2011-12-09 12:25:00       0.0     0.0     0.0     0.0    0.0    0.0
41436   2011-12-09 12:31:00       0.0     0.0     0.0     0.0    0.0    0.0
41437   2011-12-09 12:49:00       0.0     0.0     0.0     0.0    0.0    0.0
41438   2011-12-09 12:50:00       0.0     0.0     0.0     0.0    0.0    0.0

                                                  …  Quantity              \
Country Channel Islands Cyprus Czech Republic  … Singapore Spain Sweden
0                   0.0    0.0             0.0  …         0     0      0
1                   0.0    0.0             0.0  …         0     0      0
2                   0.0    0.0             0.0  …         0     0      0
3                   0.0    0.0             0.0  …         0     0      0
4                   0.0    0.0             0.0  …         0     0      0
…                   …    …            …  …       …   …    …
41434               0.0    0.0             0.0  …         0     0      0
41435               0.0    0.0             0.0  …         0     0      0
41436               0.0    0.0             0.0  …         0     0      0
41437               0.0    0.0             0.0  …         0     0      0
41438               0.0    0.0             0.0  …         0     0      0

                                                                        \
Country Switzerland Thailand USA United Arab Emirates United Kingdom
0                 0        0   0                    0            166
1                 0        0   0                    0             60
2                 0        0   0                    0            193
3                 0        0   0                    0            145
4                 0        0   0                    0            826
…             …       …  ..                  …            …
41434             0        0   0                    0             76
41435             0        0   0                    0            120
41436             0        0   0                    0            278
41437             0        0   0                    0             66
41438             0        0   0                    0              0


Country Unspecified West Indies
0                 0           0
1                 0           0
2                 0           0
3                 0           0
4                 0           0
…             …         …
41434             0           0
41435             0           0
```

```
41436              0            0
41437              0            0
41438              0            0

[41439 rows x 83 columns]
```

It has 2 leves, Price level and Quantity level

For the Price, get the mean

[ ]:
```
country_pivoted= pd.pivot_table(country_date_qty,index='date',columns='Country',
↪values=['Quantity','Price'],fill_value=0,aggfunc={'Quantity':np.sum,
↪  'Price':np.mean}).reset_index()
```

[ ]:
```
country_pivoted
```

[ ]:
```
                      date     Price                                        \
Country                          Australia Austria Bahrain Belgium Brazil Canada
0       2009-12-01 07:45:00       0.0     0.0     0.0     0.0     0.0    0.0
1       2009-12-01 07:46:00       0.0     0.0     0.0     0.0     0.0    0.0
2       2009-12-01 09:06:00       0.0     0.0     0.0     0.0     0.0    0.0
3       2009-12-01 09:08:00       0.0     0.0     0.0     0.0     0.0    0.0
4       2009-12-01 09:24:00       0.0     0.0     0.0     0.0     0.0    0.0
…                       …       …     …     …     …     …    …
41434   2011-12-09 12:23:00       0.0     0.0     0.0     0.0     0.0    0.0
41435   2011-12-09 12:25:00       0.0     0.0     0.0     0.0     0.0    0.0
41436   2011-12-09 12:31:00       0.0     0.0     0.0     0.0     0.0    0.0
41437   2011-12-09 12:49:00       0.0     0.0     0.0     0.0     0.0    0.0
41438   2011-12-09 12:50:00       0.0     0.0     0.0     0.0     0.0    0.0

                                                  …  Quantity              \
Country Channel Islands Cyprus Czech Republic  … Singapore Spain Sweden
0                    0.0    0.0            0.0  …         0     0      0
1                    0.0    0.0            0.0  …         0     0      0
2                    0.0    0.0            0.0  …         0     0      0
3                    0.0    0.0            0.0  …         0     0      0
4                    0.0    0.0            0.0  …         0     0      0
…                  …    …            …  …       …   …    …
41434                0.0    0.0            0.0  …         0     0      0
41435                0.0    0.0            0.0  …         0     0      0
41436                0.0    0.0            0.0  …         0     0      0
41437                0.0    0.0            0.0  …         0     0      0
41438                0.0    0.0            0.0  …         0     0      0

                                                                        \
Country Switzerland Thailand USA United Arab Emirates United Kingdom
```

```
0                 0        0   0                   0            166
1                 0        0   0                   0             60
2                 0        0   0                   0            193
3                 0        0   0                   0            145
4                 0        0   0                   0            826
…               …      …   ..                  …           …
41434             0        0   0                   0             76
41435             0        0   0                   0            120
41436             0        0   0                   0            278
41437             0        0   0                   0             66
41438             0        0   0                   0              0


Country Unspecified West Indies
0                 0           0
1                 0           0
2                 0           0
3                 0           0
4                 0           0
…               …         …
41434             0           0
41435             0           0
41436             0           0
41437             0           0
41438             0           0

[41439 rows x 83 columns]
```

0.11 Melt the Data

What if i want only one level, then I will connect Level 0 [Price, Quantity], Level 1 [Country names]

[ ]:
```
country_pivoted.columns
```

[ ]:
```
MultiIndex([(    'date',                    ''),
            (   'Price',            'Australia'),
            (   'Price',              'Austria'),
            (   'Price',              'Bahrain'),
            (   'Price',              'Belgium'),
            (   'Price',               'Brazil'),
            (   'Price',               'Canada'),
            (   'Price',      'Channel Islands'),
            (   'Price',               'Cyprus'),
            (   'Price',       'Czech Republic'),
            (   'Price',              'Denmark'),
            (   'Price',      'Eastern Ireland'),
            (   'Price',   'European Community'),
```

```
(   'Price',              'Finland'),
(   'Price',               'France'),
(   'Price',              'Germany'),
(   'Price',               'Greece'),
(   'Price',              'Iceland'),
(   'Price',               'Israel'),
(   'Price',                'Italy'),
(   'Price',                'Japan'),
(   'Price',                'Korea'),
(   'Price',              'Lebanon'),
(   'Price',            'Lithuania'),
(   'Price',                'Malta'),
(   'Price',          'Netherlands'),
(   'Price',              'Nigeria'),
(   'Price',               'Norway'),
(   'Price',               'Poland'),
(   'Price',             'Portugal'),
(   'Price',                  'RSA'),
(   'Price',         'Saudi Arabia'),
(   'Price',            'Singapore'),
(   'Price',                'Spain'),
(   'Price',               'Sweden'),
(   'Price',          'Switzerland'),
(   'Price',              'Thailand'),
(   'Price',                  'USA'),
(   'Price', 'United Arab Emirates'),
(   'Price',       'United Kingdom'),
(   'Price',          'Unspecified'),
(   'Price',          'West Indies'),
('Quantity',            'Australia'),
('Quantity',              'Austria'),
('Quantity',              'Bahrain'),
('Quantity',              'Belgium'),
('Quantity',               'Brazil'),
('Quantity',               'Canada'),
('Quantity',      'Channel Islands'),
('Quantity',               'Cyprus'),
('Quantity',       'Czech Republic'),
('Quantity',              'Denmark'),
('Quantity',      'Eastern Ireland'),
('Quantity',   'European Community'),
('Quantity',              'Finland'),
('Quantity',               'France'),
('Quantity',              'Germany'),
('Quantity',               'Greece'),
('Quantity',              'Iceland'),
('Quantity',               'Israel'),
```

```
            ('Quantity',                'Italy'),
            ('Quantity',                'Japan'),
            ('Quantity',                'Korea'),
            ('Quantity',              'Lebanon'),
            ('Quantity',            'Lithuania'),
            ('Quantity',                'Malta'),
            ('Quantity',          'Netherlands'),
            ('Quantity',              'Nigeria'),
            ('Quantity',               'Norway'),
            ('Quantity',               'Poland'),
            ('Quantity',             'Portugal'),
            ('Quantity',                  'RSA'),
            ('Quantity',         'Saudi Arabia'),
            ('Quantity',            'Singapore'),
            ('Quantity',                'Spain'),
            ('Quantity',               'Sweden'),
            ('Quantity',          'Switzerland'),
            ('Quantity',             'Thailand'),
            ('Quantity',                  'USA'),
            ('Quantity', 'United Arab Emirates'),
            ('Quantity',       'United Kingdom'),
            ('Quantity',          'Unspecified'),
            ('Quantity',          'West Indies')],
           names=[None, 'Country'])
```

for every level 0 and level 1 it will join them

[ ]:
```
country_pivoted.columns=country_pivoted.columns.map('_'.join)
```

[ ]:
```
country_pivoted
```

[ ]:
```
                    date_  Price_Australia  Price_Austria  Price_Bahrain  \
0     2009-12-01 07:45:00              0.0            0.0            0.0
1     2009-12-01 07:46:00              0.0            0.0            0.0
2     2009-12-01 09:06:00              0.0            0.0            0.0
3     2009-12-01 09:08:00              0.0            0.0            0.0
4     2009-12-01 09:24:00              0.0            0.0            0.0
…                     …              …              …              …
41434 2011-12-09 12:23:00              0.0            0.0            0.0
41435 2011-12-09 12:25:00              0.0            0.0            0.0
41436 2011-12-09 12:31:00              0.0            0.0            0.0
41437 2011-12-09 12:49:00              0.0            0.0            0.0
41438 2011-12-09 12:50:00              0.0            0.0            0.0

       Price_Belgium  Price_Brazil  Price_Canada  Price_Channel Islands  \
0                0.0           0.0           0.0                    0.0
1                0.0           0.0           0.0                    0.0
2                0.0           0.0           0.0                    0.0
```

```
3                0.0           0.0           0.0                     0.0
4                0.0           0.0           0.0                     0.0
…              …             …             …                       …
41434            0.0           0.0           0.0                     0.0
41435            0.0           0.0           0.0                     0.0
41436            0.0           0.0           0.0                     0.0
41437            0.0           0.0           0.0                     0.0
41438            0.0           0.0           0.0                     0.0

       Price_Cyprus  Price_Czech Republic  …  Quantity_Singapore  \
0               0.0                   0.0  …                   0
1               0.0                   0.0  …                   0
2               0.0                   0.0  …                   0
3               0.0                   0.0  …                   0
4               0.0                   0.0  …                   0
…             …                     …  …                 …
41434           0.0                   0.0  …                   0
41435           0.0                   0.0  …                   0
41436           0.0                   0.0  …                   0
41437           0.0                   0.0  …                   0
41438           0.0                   0.0  …                   0

       Quantity_Spain  Quantity_Sweden  Quantity_Switzerland  \
0                   0                0                     0
1                   0                0                     0
2                   0                0                     0
3                   0                0                     0
4                   0                0                     0
…                 …                …                   …
41434               0                0                     0
41435               0                0                     0
41436               0                0                     0
41437               0                0                     0
41438               0                0                     0

       Quantity_Thailand  Quantity_USA  Quantity_United Arab Emirates  \
0                      0             0                              0
1                      0             0                              0
2                      0             0                              0
3                      0             0                              0
4                      0             0                              0
…                    …             …                            …
41434                  0             0                              0
41435                  0             0                              0
41436                  0             0                              0
41437                  0             0                              0
41438                  0             0                              0
```

```
       Quantity_United Kingdom  Quantity_Unspecified  Quantity_West Indies
0                          166                     0                     0
1                           60                     0                     0
2                          193                     0                     0
3                          145                     0                     0
4                          826                     0                     0
…                          …                       …                     …
41434                       76                     0                     0
41435                      120                     0                     0
41436                      278                     0                     0
41437                       66                     0                     0
41438                        0                     0                     0

[41439 rows x 83 columns]
```

[ ]:
```
country_pivoted.shape
```

[ ]:
```
(41439, 83)
```

They are now in one level

To melt means to collect all the columns above into one column

[ ]:
```
pd.melt(country_pivoted,id_vars=['date_'],var_name=['Measure'])
```

[ ]:
```
                      date_               Measure  value
0       2009-12-01 07:45:00       Price_Australia    0.0
1       2009-12-01 07:46:00       Price_Australia    0.0
2       2009-12-01 09:06:00       Price_Australia    0.0
3       2009-12-01 09:08:00       Price_Australia    0.0
4       2009-12-01 09:24:00       Price_Australia    0.0
…                       …                     …    …
3397993 2011-12-09 12:23:00  Quantity_West Indies    0.0
3397994 2011-12-09 12:25:00  Quantity_West Indies    0.0
3397995 2011-12-09 12:31:00  Quantity_West Indies    0.0
3397996 2011-12-09 12:49:00  Quantity_West Indies    0.0
3397997 2011-12-09 12:50:00  Quantity_West Indies    0.0

[3397998 rows x 3 columns]
```

All the columns are collapses and combined into Measure column

Good for data combined as one variable.

[ ]:
```
pd.melt(country_pivoted,id_vars=['date_'],var_name=['Measure']).shape
```

[ ]:
```
(3397998, 3)
```

0.12 Joining

[ ]:
```
designation_data= pd.DataFrame({'name':
 ↪['mike','jonathan','Mo','Lisa','Raj','Teo'],
                                 'title':
 ↪['manager','supervisor','director','associate','assistant','sectionheead']})

age_data= pd.DataFrame({'name':['mike','jonathan','lee','Lisa','tom','Teo'],
                                 'age':['40','50','34','25','60','29']})
```

[ ]:
```
designation_data
```

[ ]:
```
       name         title
0      mike       manager
1  jonathan    supervisor
2        Mo      director
3      Lisa     associate
4       Raj     assistant
5       Teo  sectionheead
```

[ ]:
```
age_data
```

[ ]:
```
       name age
0      mike  40
1  jonathan  50
2       lee  34
3      Lisa  25
4       tom  60
5       Teo  29
```

[ ]:
```
#'left join'
pd.merge(designation_data,age_data,how='left')
```

[ ]:
```
       name         title  age
0      mike       manager   40
1  jonathan    supervisor   50
2        Mo      director  NaN
3      Lisa     associate   25
4       Raj     assistant  NaN
5       Teo  sectionheead   29
```

[ ]:
```
### full join(outer)
pd.merge(designation_data,age_data,how='outer')
```

[ ]:
```
       name         title  age
0      mike       manager   40
1  jonathan    supervisor   50
2        Mo      director  NaN
```

```
3      Lisa     associate   25
4       Raj     assistant  NaN
5       Teo  sectionheead   29
6       lee           NaN   34
7       tom           NaN   60
```

[ ]:
```
### inner join

pd.merge(designation_data,age_data,how='inner')
```

[ ]:
```
       name         title age
0      mike       manager  40
1  jonathan    supervisor  50
2      Lisa     associate  25
3       Teo  sectionheead  29
```