In [5]:
```
import pandas as pd
import numpy as np
```

In [3]:
```
df1=pd.read_csv('https://raw.githubusercontent.com/kjam/data-wrangling-pycon/maste
```

In [4]:
```
df=pd.read_csv('https://raw.githubusercontent.com/jackiekazil/data-wrangling/maste
```

In [5]:
```
df.head()
```

Out[5]:

	Indicator	PUBLISH STATES	Year	WHO region	World Bank income group	Country	Sex	Display Value	Numeric	Low	H
0	Life expectancy at birth (years)	Published	1990	Europe	High-income	Andorra	Both sexes	77	77.0	NaN	N
1	Life expectancy at birth (years)	Published	2000	Europe	High-income	Andorra	Both sexes	80	80.0	NaN	N
2	Life expectancy at age 60 (years)	Published	2012	Europe	High-income	Andorra	Female	28	28.0	NaN	N
3	Life expectancy at age 60 (years)	Published	2000	Europe	High-income	Andorra	Both sexes	23	23.0	NaN	N
4	Life expectancy at birth (years)	Published	2012	Eastern Mediterranean	High-income	United Arab Emirates	Female	78	78.0	NaN	N

In [6]:

```
df1.head()
```

Out[6]:

	STATION	STATION_NAME	DATE	PRCP	SNWD	SNOW	TMAX	TMIN	WDFG
0	GHCND:GME00111445	BERLIN TEMPELHOF GM	19310101	46	-9999	-9999	-9999	-11	-9999
1	GHCND:GME00111445	BERLIN TEMPELHOF GM	19310102	107	-9999	-9999	50	11	-9999
2	GHCND:GME00111445	BERLIN TEMPELHOF GM	19310103	-9999	-9999	-9999	28	11	-9999
3	GHCND:GME00111445	BERLIN TEMPELHOF GM	19310105	13	-9999	-9999	39	11	-9999
4	GHCND:GME00111445	BERLIN TEMPELHOF GM	19310106	-9999	-9999	-9999	0	-22	-9999

5 rows × 21 columns

1.Get the metadata from the above files.

In [61]:

```
df.dtypes
```

Out[61]:

```
Indicator                     object
PUBLISH STATES                object
CalYear                        int64
Region HWO                    object
World Bank income group       object
Country                       object
Sex                           object
Display Value                  int64
Numeric                      float64
Low                          float64
High                         float64
Comments                     float64
dtype: object
```

```
In [62]: df1.dtypes
```

```
Out[62]: STATION         object
         STATION_NAME    object
         DATE             int64
         PRCP             int64
         SNWD             int64
         SNOW             int64
         TMAX             int64
         TMIN             int64
         WDFG             int64
         PGTM             int64
         WSFG             int64
         WT09             int64
         WT07             int64
         WT01             int64
         WT06             int64
         WT05             int64
         WT04             int64
         WT16             int64
         WT08             int64
         WT18             int64
         WT03             int64
         dtype: object
```

2.Get the row names from the above files

```
In [21]: df.index
```

```
Out[21]: RangeIndex(start=0, stop=4656, step=1)
```

```
In [24]: (df.index).values
```

```
Out[24]: array([   0,    1,    2, ..., 4653, 4654, 4655], dtype=int64)
```

```
In [25]: (df1.index).values
```

```
Out[25]: array([     0,      1,      2, ..., 117205, 117206, 117207], dtype=int64)
```

3. Change the column name from any of the above file.

```
In [26]: df.columns.values[3]
```

```
Out[26]: 'WHO region'
```

```
In [27]: df.columns.values[3]="region HOW"
```

```
In [28]: df.columns.values[3]
```

```
Out[28]: 'region HOW'
```

4. Change the Column name from any of the above file and store the cahnges made

permanentely.

In [30]:
```
df.rename(columns=lambda x: x.replace('Year','year'),inplace=True)
```

In [31]:
```
df.head()
```

Out[31]:

	Indicator	PUBLISH STATES	year	region HOW	World Bank income group	Country	Sex	Display Value	Numeric	Low	H
0	Life expectancy at birth (years)	Published	1990	Europe	High-income	Andorra	Both sexes	77	77.0	NaN	N
1	Life expectancy at birth (years)	Published	2000	Europe	High-income	Andorra	Both sexes	80	80.0	NaN	N
2	Life expectancy at age 60 (years)	Published	2012	Europe	High-income	Andorra	Female	28	28.0	NaN	N
3	Life expectancy at age 60 (years)	Published	2000	Europe	High-income	Andorra	Both sexes	23	23.0	NaN	N
4	Life expectancy at birth (years)	Published	2012	Eastern Mediterranean	High-income	United Arab Emirates	Female	78	78.0	NaN	N

In [32]:
```
df.columns.values[[2,3]]=['CalYear','Region HWO']
```

```
In [33]: df.head()
```

Out[33]:

	Indicator	PUBLISH STATES	CalYear	Region HWO	World Bank income group	Country	Sex	Display Value	Numeric	Low
0	Life expectancy at birth (years)	Published	1990	Europe	High-income	Andorra	Both sexes	77	77.0	NaN
1	Life expectancy at birth (years)	Published	2000	Europe	High-income	Andorra	Both sexes	80	80.0	NaN
2	Life expectancy at age 60 (years)	Published	2012	Europe	High-income	Andorra	Female	28	28.0	NaN
3	Life expectancy at age 60 (years)	Published	2000	Europe	High-income	Andorra	Both sexes	23	23.0	NaN
4	Life expectancy at birth (years)	Published	2012	Eastern Mediterranean	High-income	United Arab Emirates	Female	78	78.0	NaN

In [36]:
```
df.head()
```

Out[36]:

	Indicator	PUBLISH STATES	CalYear	Region HWO	World Bank income group	Country	Sex	Display Value	Numeric	Low
0	Life expectancy at birth (years)	Published	1990	Europe	High-income	Andorra	Both sexes	77	77.0	NaN
1	Life expectancy at birth (years)	Published	2000	Europe	High-income	Andorra	Both sexes	80	80.0	NaN
2	Life expectancy at age 60 (years)	Published	2012	Europe	High-income	Andorra	Female	28	28.0	NaN
3	Life expectancy at age 60 (years)	Published	2000	Europe	High-income	Andorra	Both sexes	23	23.0	NaN
4	Life expectancy at birth (years)	Published	2012	Eastern Mediterranean	High-income	United Arab Emirates	Female	78	78.0	NaN

In [39]:
```
df.sort_values(['Display Value'],ascending=True)
```

Out[39]:

	Indicator	PUBLISH STATES	CalYear	Region HWO	World Bank income group	Country	Sex	Display Value	Numeri
265	Life expectancy at age 60 (years)	Published	2000	Africa	Low-income	Sierra Leone	Male	11	11.(
1476	Life expectancy at age 60 (years)	Published	1990	Africa	Low-income	Sierra Leone	Female	11	11.(
1474	Life expectancy at age 60 (years)	Published	1990	Africa	Low-income	Sierra Leone	Male	11	11.(
	Life expectancy				Low-	Sierra			

7. Arrange multiple column values in ascending order.

In [40]:
```
df.sort_values(['Display Value','Sex'],ascending=True)
```

Out[40]:

	Indicator	PUBLISH STATES	CalYear	Region HWO	World Bank income group	Country	Sex	Display Value	Numeri
3071	Life expectancy at age 60 (years)	Published	1990	Africa	Low-income	Sierra Leone	Both sexes	11	11.
3072	Life expectancy at age 60 (years)	Published	2000	Africa	Low-income	Sierra Leone	Both sexes	11	11.
1476	Life expectancy at age 60 (years)	Published	1990	Africa	Low-income	Sierra Leone	Female	11	11.
	Life expectancy				Low-	Sierra			

8. Make country as the first Column of the datafarmae

9.Get the column array using a variable

In [68]:
```
get_Column_Sex=df["Sex"].values
get_Column_Sex
```

Out[68]:
```
array(['Both sexes', 'Both sexes', 'Female', ..., 'Male', 'Both sexes',
       'Female'], dtype=object)
```

10. Get the subset rows 11,24,37

In [70]:
```
df.iloc[[11,24,37],:]
```

Out[70]:

	Indicator	PUBLISH STATES	CalYear	Region HWO	World Bank income group	Country	Sex	Display Value	Numeric	Low
11	Life expectancy at birth (years)	Published	2012	Europe	High-income	Austria	Female	83	83.0	NaN
24	Life expectancy at age 60 (years)	Published	2012	Western Pacific	High-income	Brunei Darussalam	Female	21	21.0	NaN
37	Life expectancy at age 60 (years)	Published	2012	Europe	High-income	Cyprus	Female	26	26.0	NaN

11. Get the subset rows excluding 5,12,23, and 56

In [79]:
```
new_df=df[(df.index !=5) & (df.index !=12) & (df.index !=23) & (df.index !=56)]
```

In [80]:
```
new_df.head(6)
```

Out[80]:

	Indicator	PUBLISH STATES	CalYear	Region HWO	World Bank income group	Country	Sex	Display Value	Numeric	Low
0	Life expectancy at birth (years)	Published	1990	Europe	High-income	Andorra	Both sexes	77	77.0	NaN
1	Life expectancy at birth (years)	Published	2000	Europe	High-income	Andorra	Both sexes	80	80.0	NaN
2	Life expectancy at age 60 (years)	Published	2012	Europe	High-income	Andorra	Female	28	28.0	NaN
3	Life expectancy at age 60 (years)	Published	2000	Europe	High-income	Andorra	Both sexes	23	23.0	NaN
4	Life expectancy at birth (years)	Published	2012	Eastern Mediterranean	High-income	United Arab Emirates	Female	78	78.0	NaN
6	Life expectancy at age 60 (years)	Published	1990	Americas	High-income	Antigua and Barbuda	Male	17	17.0	NaN

12.Join users to transactions , keeping all rows from transactions and only matching rows from users(left join)

In [88]:
```
users=pd.read_csv('https://raw.githubusercontent.com/ben519/DataWrangling/master/D
sessions=pd.read_csv('https://raw.githubusercontent.com/ben519/DataWrangling/maste
products=pd.read_csv('https://raw.githubusercontent.com/ben519/DataWrangling/maste
transactions=pd.read_csv('https://raw.githubusercontent.com/ben519/DataWrangling/m
```

In [89]:
```
users.head()
sessions.head()
products.head()
transactions.head()
```

Out[89]:

	TransactionID	TransactionDate	UserID	ProductID	Quantity
0	1	2010-08-21	7.0	2	1
1	2	2011-05-26	3.0	4	1
2	3	2011-06-16	3.0	3	1
3	4	2012-08-26	1.0	2	3
4	5	2013-06-06	2.0	4	1

In [91]:
```
users.head()
```

Out[91]:

	UserID	User	Gender	Registered	Cancelled
0	1	Charles	male	2012-12-21	NaN
1	2	Pedro	male	2010-08-01	2010-08-08
2	3	Caroline	female	2012-10-23	2016-06-07
3	4	Brielle	female	2013-07-17	NaN
4	5	Benjamin	male	2010-11-25	NaN

In [6]:
```
import numpy as np
import pandas as pd
from pandas import DataFrame, Series
import sqlite3 as db
```

In [7]:
```
!pip install pandasql
```

```
Requirement already satisfied: pandasql in c:\users\mgirm\anaconda3\lib\site-p
ackages (0.7.3)
Requirement already satisfied: pandas in c:\users\mgirm\anaconda3\lib\site-pac
kages (from pandasql) (0.23.4)
Requirement already satisfied: sqlalchemy in c:\users\mgirm\anaconda3\lib\site
-packages (from pandasql) (1.2.11)
Requirement already satisfied: numpy in c:\users\mgirm\anaconda3\lib\site-pack
ages (from pandasql) (1.15.1)
Requirement already satisfied: python-dateutil>=2.5.0 in c:\users\mgirm\anacon
da3\lib\site-packages (from pandas->pandasql) (2.7.3)
Requirement already satisfied: pytz>=2011k in c:\users\mgirm\anaconda3\lib\sit
e-packages (from pandas->pandasql) (2018.5)
Requirement already satisfied: six>=1.5 in c:\users\mgirm\anaconda3\lib\site-p
ackages (from python-dateutil>=2.5.0->pandas->pandasql) (1.11.0)
```

In [8]:
```
!python -m pip install --upgrade pip
```

```
Requirement already up-to-date: pip in c:\users\mgirm\anaconda3\lib\site-packa
ges (18.1)
```

```
In [12]: pysqldf = lambda q: sqldf(q, globals())
```

```
In [13]: q="""select * from transactions t left join users u on t.userid=u.userid;"""
```

13. Which transaction have a UserID not in users ?

```
In [151]: df.loc[df.User.isnull(),'TransactionID']
```

```
Out[151]: 0    1
          7    8
          8    9
          Name: TransactionID, dtype: int64
```

14. Join users to transaction , keeping only rows from transactions and users that match via UserID.

```
In [160]: pysqldf = lambda p: sqldf(p, globals())
```

```
In [153]: p="""select * from transactions t inner join users u on t.userid=u.userid;"""
```

```
In [154]: df=pysqldf(p)
          df
```

Out[154]:

	TransactionID	TransactionDate	UserID	ProductID	Quantity	UserID	User	Gender	Registere
0	2	2011-05-26	3.0	4	1	3	Caroline	female	2012-10-2
1	3	2011-06-16	3.0	3	1	3	Caroline	female	2012-10-2
2	4	2012-08-26	1.0	2	3	1	Charles	male	2012-12-2
3	5	2013-06-06	2.0	4	1	2	Pedro	male	2010-08-0
4	6	2013-12-23	2.0	5	6	2	Pedro	male	2010-08-0
5	7	2013-12-30	3.0	4	1	3	Caroline	female	2012-10-2
6	10	2016-05-08	3.0	4	4	3	Caroline	female	2012-10-2

15. Join users to transaction,diplaying all matchingrows AND all non-matching rows

```
In [ ]:
```