Time Methods

Python Datetime Review

Basic Python outside of Pandas contains a datetime library:

In [1]:

```
import numpy as np
import pandas as pd
from datetime import datetime
```

In [4]:

```
# To illustrate the order of arguments
my_year = 2017
my_month = 1
my_day = 2
my_hour = 13
my_minute = 30
my_second = 15
```

In [5]:

```
# January 2nd, 2017
my_date = datetime(my_year,my_month,my_day)
```

In [6]:

```
# Defaults to 0:00
my_date
```

Out[6]:

```
datetime.datetime(2017, 1, 2, 0, 0)
```

In [8]:

```
# January 2nd, 2017 at 13:30:15
my_date_time = datetime(my_year,my_month,my_day,my_hour,my_minute,my_second)
my_date_time
```

Out[8]:

```
datetime.datetime(2017, 1, 2, 13, 30, 15)
```

You can grab any part of the datetime object you want

In [9]:

```
my_date.year
```

Out[9]:

```
2017
```

In [12]:

```
my_date_time.hour
```

Out[12]:

```
13
```

Pandas

Pandas

Converting to datetime

Often when data sets are stored, the time component may be a string. Pandas easily converts strings to datetime objects.

In [13]:

```
myser = pd.Series(['Nov 3, 2000', '2000-01-01', None])
```

In [14]:

```
myser
```

Out[14]:

```
0    Nov 3, 2000
1     2000-01-01
2           None
dtype: object
```

In [16]:

```
myser[0]
```

Out[16]:

```
'Nov 3, 2000'
```

pd.to_datetime()

https://pandas.pydata.org/pandas-docs/stable/user_guide/timeseries.html#converting-to-timestamps

In [18]:

```
pd.to_datetime(myser)
```

Out[18]:

```
0   2000-11-03
1   2000-01-01
2          NaT
dtype: datetime64[ns]
```

In [19]:

```
pd.to_datetime(myser)[0]
```

Out[19]:

```
Timestamp('2000-11-03 00:00:00')
```

In [20]:

```
# Here we mention time that have 31 as date so python can easily undestand which one is d
ate and which one is month
obvi_euro_date = '31-12-2000'
```

In [21]:

```
pd.to_datetime(obvi_euro_date)
```

```
C:\Users\Chromsy\AppData\Local\Temp\ipykernel_2436\163700324.py:1: UserWarning: Parsing d
ates in DD/MM/YYYY format when dayfirst=False (the default) was specified. This may lead
to inconsistently parsed dates! Specify a format to ensure consistent parsing.
  pd.to_datetime(obvi_euro_date)
```

Out[21]:

```
Timestamp('2000-12-31 00:00:00')
```

In [22]:

```
# 10th of Dec OR 12th of October?
# We may need to tell pandas
euro_date = '10-12-2000'
```

In [24]:

```
pd.to_datetime(euro_date)  # Here python made a guess but we can fix that if not correct
```

Out[24]:

```
Timestamp('2000-10-12 00:00:00')
```

In [26]:

```
pd.to_datetime(euro_date,dayfirst=True) # Here we set day is first
```

Out[26]:

```
Timestamp('2000-12-10 00:00:00')
```

Custom Time String Formatting

Sometimes dates can have a non standard format, luckily you can always specify to pandas the format. You should also note this could speed up the conversion, so it may be worth doing even if pandas can parse on its own.

A full table of codes can be found here: https://docs.python.org/3/library/datetime.html#strftime-and-strptime-format-codes

In [28]:

```
style_date = '12--Dec--2000' # Here datin random style
```

In [29]:

```
pd.to_datetime(style_date, format='%d--%b--%Y') # check above link to know these codes
```

Out[29]:

```
Timestamp('2000-12-12 00:00:00')
```

In [30]:

```
strange_date = '12th of Dec 2000'
```

In [32]:

```
pd.to_datetime(strange_date) # It can understand by its own
```

Out[32]:

```
Timestamp('2000-12-12 00:00:00')
```

Data

Retail Sales: Beer, Wine, and Liquor Stores

Units: Millions of Dollars, Not Seasonally Adjusted

Frequency: Monthly

U.S. Census Bureau, Retail Sales: Beer, Wine, and Liquor Stores [MRTSSM4453USN], retrieved from FRED, Federal Reserve Bank of St. Louis; https://fred.stlouisfed.org/series/MRTSSM4453USN, July 2, 2020.

In [34]:

In [34]:

```
sales = pd.read_csv("D:\\Study\\Programming\\python\\Python course from udemy\\[GigaCourse.Com] Udemy - 2022 Python for Machine Learning & Data Science Masterclass\\01 - Introduction to Course\\1UNZIP-FOR-NOTEBOOKS-FINAL\\03-Pandas\\RetailSales_BeerWineLiquor.csv")
```

In [35]:

```
sales
```

Out[35]:

	DATE	MRTSSM4453USN
0	1992-01-01	1509
1	1992-02-01	1541
2	1992-03-01	1597
3	1992-04-01	1675
4	1992-05-01	1822
...	...	...
335	2019-12-01	6630
336	2020-01-01	4388
337	2020-02-01	4533
338	2020-03-01	5562
339	2020-04-01	5207

340 rows × 2 columns

In [36]:

```
sales.iloc[0]['DATE']
```

Out[36]:

```
'1992-01-01'
```

In [39]:

```
type(sales.iloc[0]['DATE'])  # Here we see that dates are in srings
```

Out[39]:

```
str
```

In [40]:

```
sales['DATE'] = pd.to_datetime(sales['DATE'])
```

In [41]:

```
sales
```

Out[41]:

	DATE	MRTSSM4453USN
0	1992-01-01	1509
1	1992-02-01	1541
2	1992-03-01	1597
3	1992-04-01	1675
4	1992-05-01	1822
...	...	...
335	2019-12-01	6630

	DATE	MRTSSM4453USN
336	2020-01-01	4388
337	2020-02-01	4533
338	2020-03-01	5562
339	2020-04-01	5207

340 rows × 2 columns

In [42]:

```
sales.iloc[0]['DATE']
```

Out[42]:

```
Timestamp('1992-01-01 00:00:00')
```

In [43]:

```
type(sales.iloc[0]['DATE'])
```

Out[43]:

```
pandas._libs.tslibs.timestamps.Timestamp
```

Attempt to Parse Dates Automatically

parse_dates - bool or list of int or names or list of lists or dict, default False The behavior is as follows:

```
boolean. If True -> try parsing the index.

list of int or names. e.g. If [1, 2, 3] -> try parsing columns 1, 2, 3 each as a se
parate date column.

list of lists. e.g. If [[1, 3]] -> combine columns 1 and 3 and parse as a single da
te column.

dict, e.g. {'foo' : [1, 3]} -> parse columns 1, 3 as date and call result 'foo'

If a column or index cannot be represented as an array of datetimes, say because of
an unparseable value or a mixture of timezones, the column or index will be returne
d unaltered as an object data type. For non-standard datetime parsing, use pd.to_da
tetime after pd.read_csv. To parse an index or column with a mixture of timezones,
specify date_parser to be a partially-applied pandas.to_datetime() with utc=True.
See Parsing a CSV with mixed timezones for more.
```

In [45]:

```
# Parse Column at Index 0 as Datetime
sales = pd.read_csv("D:\\Study\\Programming\\python\\Python course from udemy\\[GigaCour
se.Com] Udemy - 2022 Python for Machine Learning & Data Science Masterclass\\01 - Introdu
ction to Course\\1UNZIP-FOR-NOTEBOOKS-FINAL\\03-Pandas\\RetailSales_BeerWineLiquor.csv"
                    ,parse_dates=[0])
```

In [46]:

```
sales
```

Out[46]:

	DATE	MRTSSM4453USN
0	1992-01-01	1509
1	1992-02-01	1541
2	1992-03-01	1597

	DATE	MRTSSM4453USN
3	1992-04-01	1675
4	1992-05-01	1822
...	...	...
335	2019-12-01	6630
336	2020-01-01	4388
337	2020-02-01	4533
338	2020-03-01	5562
339	2020-04-01	5207

340 rows × 2 columns

In [47]:

```
type(sales.iloc[0]['DATE'])
```

Out[47]:

```
pandas._libs.tslibs.timestamps.Timestamp
```

Resample

A common operation with time series data is resampling based on the time series index. Let's see how to use the resample() method. [reference]

In [49]:

```
# Our index
sales.index
```

Out[49]:

```
RangeIndex(start=0, stop=340, step=1)
```

In [50]:

```
# Reset DATE to index
sales = sales.set_index('DATE')
```

In [51]:

```
sales
```

Out[51]:

	MRTSSM4453USN
DATE	
1992-01-01	1509
1992-02-01	1541
1992-03-01	1597
1992-04-01	1675
1992-05-01	1822
...	...
2019-12-01	6630
2020-01-01	4388
2020-02-01	4533
2020-03-01	5562
2020-04-01	5207

	MRTSSM4453USN
DATE	

340 rows × 1 columns

When calling `.resample()` you first need to pass in a **rule** parameter, then you need to call some sort of aggregation function.

The **rule** parameter describes the frequency with which to apply the aggregation function (daily, monthly, yearly, etc.)
It is passed in using an "offset alias" - refer to the table below. [reference]

The aggregation function is needed because, due to resampling, we need some sort of mathematical rule to join the rows (mean, sum, count, etc.)

TIME SERIES OFFSET ALIASES

ALIAS	DESCRIPTION
B	business day frequency
C	custom business day frequency (experimental)
D	calendar day frequency
W	weekly frequency
M	month end frequency
SM	semi-month end frequency (15th and end of month)
BM	business month end frequency
CBM	custom business month end frequency
MS	month start frequency
SMS	semi-month start frequency (1st and 15th)
BMS	business month start frequency
CBMS	custom business month start frequency
Q	quarter end frequency
	intentionally left blank

ALIAS	DESCRIPTION
BQ	business quarter endfrequency
QS	quarter start frequency
BQS	business quarter start frequency
A	year end frequency
BA	business year end frequency
AS	year start frequency
BAS	business year start frequency
BH	business hour frequency
H	hourly frequency
T, min	minutely frequency
S	secondly frequency
L, ms	milliseconds
U, us	microseconds
N	nanoseconds

In [53]:

```
# Yearly Means # Here ruel A can see in above list
sales.resample(rule='A').mean()
```

Out[53]:

	MRTSSM4453USN
DATE	
1992-12-31	1807.250000
1993-12-31	1794.833333
1994-12-31	1841.750000
1995-12-31	1833.916667
1996-12-31	1929.750000
1997-12-31	2006.750000
1998-12-31	2115.166667
1999-12-31	2206.333333
2000-12-31	2375.583333
2001-12-31	2468.416667
2002-12-31	2491.166667
2003-12-31	2539.083333

DATE	MRTSSM4453USN
2004-12-31	2682.416667
2005-12-31	2797.250000
2006-12-31	3001.333333
2007-12-31	3177.333333
2008-12-31	3292.000000
2009-12-31	3353.750000
2010-12-31	3450.083333
2011-12-31	3532.666667
2012-12-31	3697.083333
2013-12-31	3839.666667
2014-12-31	4023.833333
2015-12-31	4212.500000
2016-12-31	4434.416667
2017-12-31	4602.666667
2018-12-31	4830.666667
2019-12-31	4972.750000
2020-12-31	4922.500000

Resampling rule 'A' takes all of the data points in a given year, applies the aggregation function (in this case we calculate the mean), and reports the result as the last day of that year. Note 2020 in this data set was not complete.

.dt Method Calls

Once a column or index is ina datetime format, you can call a variety of methods off of the .dt library inside pandas:

https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.Series.dt.html

In [55]:

```
sales = sales.reset_index()
sales
```

Out[55]:

	index	DATE	MRTSSM4453USN
0	0	1992-01-01	1509
1	1	1992-02-01	1541
2	2	1992-03-01	1597
3	3	1992-04-01	1675
4	4	1992-05-01	1822
...	...	...	...
335	335	2019-12-01	6630
336	336	2020-01-01	4388
337	337	2020-02-01	4533
338	338	2020-03-01	5562
339	339	2020-04-01	5207

340 rows × 3 columns

In [56]:

```
help(sales['DATE'].dt)
```

In [57]:

```
sales['DATE'].dt.month
```

Out[57]:

```
0       1
1       2
2       3
3       4
4       5
       ..
335    12
336     1
337     2
338     3
339     4
Name: DATE, Length: 340, dtype: int64
```

In [58]:

```
sales['DATE'].dt.is_leap_year
```

Out[58]:

```
0       True
1       True
2       True
3       True
4       True
       ...
335    False
336     True
337     True
338     True
339     True
Name: DATE, Length: 340, dtype: bool
```

Inputs and Outputs

NOTE: Typically we will just be either reading csv files directly or using pandas-datareader to pull data from the web. Consider this lecture just a quick overview of what is possible with pandas (we won't be working with SQL or Excel files in this course)

Data Input and Output

This notebook is the reference code for getting input and output, pandas can read a variety of file types using its pd.read_ methods. Let's take a look at the most common data types:

Check out the references here!

This is the best online resource for how to read/write to a variety of data sources!

https://pandas.pydata.org/pandas-docs/stable/user_guide/io.html

Format Type	Data Description	Reader	Writer
text	CSV	read_csv	to_csv
text	JSON	read_json	to_json
text	HTML	read_html	to_html
text	Local clipboard	read_clipboard	to_clipboard
binary	MS Excel	read_excel	to_excel
binary	OpenDocument	read_excel	
binary	HDF5 Format	read_hdf	to_hdf
binary	Feather Format	read_feather	to_feather
binary	Parquet Format	read_parquet	to_parquet
binary	Msgpack	read_msgpack	to_msgpack
binary	Stata	read_stata	to_stata
binary	SAS	read_sas	
binary	Python Pickle Format	read_pickle	to_pickle
SQL	SQL	read_sql	to_sql
SQL	Google Big Query	read_gbq	to_gbq

Reading in a CSV

Comma Separated Values files are text files that use commas as field delimeters.
Unless you're running the virtual environment included with the course, you may need to install `xlrd` **and** `openpyxl`**.**
In your terminal/command prompt run:

```
conda install xlrd
conda install openpyxl
```

Then restart Jupyter Notebook. (or use pip install if you aren't using the Anaconda Distribution)

Understanding File Paths

You have two options when reading a file with pandas:

1. **If your .py file or .ipynb notebook is located in the exact same folder location as the .csv file you want to read, simply pass in the file name as a string, for example:**

    ```
    df = pd.read_csv('some_file.csv')
    ```

2. **Pass in the entire file path if you are located in a different directory. The file path must be 100% correct in order for this to work. For example:**

    ```
    df = pd.read_csv("C:\\Users\\myself\\files\\some_file.csv")
    ```

Print your current directory file path with pwd

In [60]:

```
pwd
```

Out[60]:

```
'C:\\Users\\Chromsy'
```

List the files in your current directory with ls

In [61]:

```
ls
```

```
 Volume in drive C has no label.
 Volume Serial Number is 246B-60C4

 Directory of C:\Users\Chromsy

24-12-2022  01:08    <DIR>          .
24-12-2022  01:08    <DIR>          ..
24-12-2022  01:09    <DIR>          .conda
11-12-2022  12:41                25 .condarc
11-12-2022  12:41    <DIR>          .continuum
23-12-2022  23:22    <DIR>          .ipynb_checkpoints
11-12-2022  20:21    <DIR>          .ipython
11-12-2022  12:40    <DIR>          .jupyter
28-11-2022  20:19    <DIR>          .skiko
30-04-2022  21:57             6,881 -1.14-windows.xml
30-04-2022  14:42    <DIR>          3D Objects
30-04-2022  14:42    <DIR>          Contacts
12-12-2022  20:50    <DIR>          Desktop
04-12-2022  02:23    <DIR>          Documents
24-12-2022  01:06    <DIR>          Downloads
13-12-2022  13:30            29,705 Downloads.ipynb
30-04-2022  14:42    <DIR>          Favorites
30-04-2022  14:42    <DIR>          Links
30-04-2022  14:42    <DIR>          Music
14-12-2022  17:09            43,279 NumPy.ipynb
30-04-2022  14:46    <DIR>          OneDrive
24-12-2022  01:08            52,082 Pandas  Input and Output.ipynb
22-12-2022  00:44            72,444 Pandas Combining DataFrames , Text Methods.ipynb
19-12-2022  02:04           252,099 Pandas Conditional Formatting ,Useful methods.ipynb
21-12-2022  01:09           249,801 Pandas Missing values, Groupby Operations and Multi-l
evel Index.ipynb
16-12-2022  19:03           169,112 Pandas Series and dataframe.ipynb
20-12-2022  18:47    <DIR>          Pictures
30-04-2022  14:42    <DIR>          Saved Games
30-04-2022  14:44    <DIR>          Searches
13-12-2022  01:16            18,752 tips.csv
29-05-2022  16:20    <DIR>          Tracing
13-12-2022  01:17            29,705 U Basics.ipynb
09-05-2022  11:55    <DIR>          Videos
              11 File(s)        923,885 bytes
              22 Dir(s)   1,960,468,480 bytes free
```

NOTE! Common confusion point! Take note that all read input methods are called directly from pandas with pd.read , *all output methods are called directly off the dataframe with df.to*

CSV Input

In [63]:

```
df= pd.read_csv("D:\\Study\\Programming\python\Python course from udemy\\[GigaCourse.Com
] Udemy - 2022 Python for Machine Learning & Data Science Masterclass\\01 - Introduction
to Course\\1UNZIP-FOR-NOTEBOOKS-FINAL\\03-Pandas\\example.csv")
df
```

Out[63]:

	a	b	c	d
0	0	1	2	3
1	4	5	6	7
2	8	9	10	11
3	12	13	14	15

In [68]:

```
df = pd.read_csv("D:\\Study\\Programming\python\Python course from udemy\\[GigaCourse.Co
m] Udemy - 2022 Python for Machine Learning & Data Science Masterclass\\01 - Introduction
to Course\\1UNZIP-FOR-NOTEBOOKS-FINAL\\03-Pandas\\example.csv"
,header=None)
df # Here we remove as a b c d as  headers now this Dataframe donr have any header
```

Out[68]:

	0	1	2	3
0	a	b	c	d
1	0	1	2	3
2	4	5	6	7
3	8	9	10	11
4	12	13	14	15

In [69]:

```
df = pd.read_csv("D:\\Study\\Programming\python\Python course from udemy\\[GigaCourse.Co
m] Udemy - 2022 Python for Machine Learning & Data Science Masterclass\\01 - Introduction
to Course\\1UNZIP-FOR-NOTEBOOKS-FINAL\\03-Pandas\\example.csv"
,index_col=0)
df # Here we set index as 0 column we cansue it by set by set_index(df['a'])
```

Out[69]:

	b	c	d
a			
0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

CSV Output

Set index=False if you do not want to save the index , otherwise it will add a new column to the .csv file that includes your index and call it "Unnamed: 0" if your index did not have a name. If you do want to save your index, simply set it to True (the default value).

In [71]:

```
df.to_csv('D:\\Study\\new_file.csv') # Here we have write file as well with formate
```

In [72]:

```
df.to_csv('D:\\Study\\new_file.csv',index=False) # Here we have write file as well with f
ormate , Here index is false means
                                                  # it would not take any extra columns f
or index
```

HTML

Pandas can read table tabs off of HTML. This only works if your firewall isn't blocking pandas from accessing the internet!

Unless you're running the virtual environment included with the course, you may need to install `lxml`, `htmllib5`, **and** `BeautifulSoup4`.
In your terminal/command prompt run:

```
conda install lxml

or

pip install lxml
```

Then restart Jupyter Notebook (you may need to restart your computer). (or use pip install if you aren't using the Anaconda Distribution)

In [2]:

```
url = "https://en.wikipedia.org/wiki/World_population"
```

In [5]:

```
tables = pd.read_html(url) # here it shows all tables from the page
tables
```

Out[5]:

```
[      Population          1      2      3      4      5      6      7      8      9  \
 0           Year       1804   1930   1960   1974   1987   1999   2011   2022   2037
 1  Years elapsed   200,000+    126     30     14     13     12     12     11     15

      10
 0  2057
 1    20  ,
                                                       #  \
 0                                                     1
 1                                                     2
 2                                                     3
 3                                                     4
 4                                                     5
 5                                                     6
 6                                                     7
 7                                                     8
 8                                                     9
 9                                                    10
 10                                                  NaN
 11  Notes: .mw-parser-output .reflist{font-size:90...

                                  Most populous countries  \
 0                                               China[B]
 1                                                  India
 2                                          United States
 3                                              Indonesia
 4                                               Pakistan
 5                                                 Brazil
 6                                                Nigeria
 7                                             Bangladesh
 8                                                 Russia
 9                                                 Mexico
 10                                           World total
 11  Notes: .mw-parser-output .reflist{font-size:90...

                                                    2000  \
 0                                                  1270
 1                                                  1053
```

```
2                                              283
3                                              212
4                                              136
5                                              176
6                                              123
7                                              131
8                                              146
9                                              103
10                                            6127
11  Notes: .mw-parser-output .reflist{font-size:90...

                                              2015  \
0                                             1376
1                                             1311
2                                              322
3                                              258
4                                              208
5                                              206
6                                              182
7                                              161
8                                              146
9                                              127
10                                            7349
11  Notes: .mw-parser-output .reflist{font-size:90...

                                           2030[A]
0                                             1416
1                                             1528
2                                              356
3                                              295
4                                              245
5                                              228
6                                              263
7                                              186
8                                              149
9                                              148
10                                            8501
11  Notes: .mw-parser-output .reflist{font-size:90...  ,
                    Region Density(inhabitants/km2) Population(millions)  \
0                     Asia                    104.1                 4641
1                   Africa                     44.4                 1340
2                   Europe                     73.4                  747
3            Latin America                     24.1                  653
4  Northern America[note 2]                    14.9                  368
5                  Oceania                        5                   42
6               Antarctica                       ~0             0.004[91]

                              Most populous country  \
0                     1,411,778,000 - China[note 1]
1                           0,211,401,000 - Nigeria
2  0,146,171,000 - Russia, approx. 110 million in...
3                            0,214,103,000 - Brazil
4                     0,332,909,000 - United States
5                         0,025,917,000 - Australia
6                                      N/A[note 3]

             Most populous city (metropolitan area)
0  13,515,000 - Tokyo Metropolis(37,400,000 - Gre...
1     09,500,000 - Cairo(20,076,000 - Greater Cairo)
2  13,200,000 - Moscow(20,004,000 - Moscow metrop...
3  12,252,000 - São Paulo City(21,650,000 - São P...
4  08,804,000 - New York City(23,582,649 - New Yo...
5                              05,367,000 - Sydney
6                     00,001,258 - McMurdo Station  ,
   Rank Country / Dependency  Population  Percentage of the world  \
0     1                China  1412600000                      NaN
1     2                India  1373761000                      NaN
2     3        United States   333472984                      NaN
3     4            Indonesia   275773800                      NaN
4     5             Pakistan   229488994                      NaN
5     6              Nigeria   216746934                      NaN
6     7               Brazil   215552699                      NaN
```

```
7    8          Bangladesh  168220000                    NaN
8    9              Russia  147190000                    NaN
9   10              Mexico  128271248                    NaN

         Date Source (official or from the United Nations)
0  31 Dec 2021                National annual estimate[93]
1   1 Mar 2022                Annual national estimate[94]
2  23 Dec 2022               National population clock[95]
3   1 Jul 2022                National annual estimate[96]
4   1 Jul 2022                           UN projection[97]
5   1 Jul 2022                           UN projection[97]
6  23 Dec 2022               National population clock[98]
7   1 Jul 2020              Annual Population Estimate[99]
8   1 Oct 2021        2021 preliminary census results[100]
9  31 Mar 2022             National quarterly estimate[101]  ,
   Rank          Country  Population  Area(km2)  Density(pop/km2)
0     1        Singapore     5921231        719              8235
1     2       Bangladesh   165650475     148460              1116
2     3  Palestine[103]     5223000       6025               867
3     4          Lebanon     5296814      10400               509
4     5           Taiwan    23580712      35980               655
5     6      South Korea    51844834      99720               520
6     7           Rwanda    13173730      26338               500
7     8           Israel     8914885      21937               406
8     9            Haiti    11334637      27750               408
9    10      Netherlands    17400824      41543               419,
   Rank          Country  Population  Area(km2)  Density(pop/km2)  \
0     1            India  1389637446    3287263               423
1     2         Pakistan   242923845     796095               305
2     3       Bangladesh   165650475     148460              1116
3     4            Japan   124214766     377915               329
4     5      Philippines   114597229     300000               382
5     6          Vietnam   103808319     331210               313
6     7  United Kingdom    67791400     243610               278
7     8      South Korea    51844834      99720               520
8     9           Taiwan    23580712      35980               655
9    10        Sri Lanka    23187516      65610               353

  Population trend[citation needed]
0                           Growing
1                  Rapidly growing
2                  Rapidly growing
3                     Declining[104]
4                           Growing
5                           Growing
6                           Growing
7                            Steady
8                            Steady
9                           Growing  ,
    Year  Population Yearly growth           Density(pop/km2)  \
    Year  Population             %    Number Density(pop/km2)
0   1951  2584034261         1.88%  47603112               17
1   1952  2630861562         1.81%  46827301               18
2   1953  2677608960         1.78%  46747398               18
3   1954  2724846741         1.76%  47237781               18
4   1955  2773019936         1.77%  48173195               19
..   ...         ...           ...       ...              ...
65  2016  7464022000         1.14%  84225000               50
66  2017  7547859000         1.12%  83837000               51
67  2018  7631091000         1.10%  83232000               51
68  2019  7713468000         1.08%  82377000               52
69  2020  7795000000         1.05%  81331000               52

   Urban population
             Number    %
0         775067697  30%
1         799282533  30%
2         824289989  31%
3         850179106  31%
4         877008842  32%
..              ...  ...
65       4060653000  54%
```

```
66        4140189000  55%
67        4219817000  55%
68        4299439000  56%
69        4378900000  56%

[70 rows x 7 columns],
                     Region  1500  1600  1700  1750  1800  1850  1900  1950  \
0                     World   585   660   710   791   978  1262  1650  2521
1                    Africa    86   114   106   106   107   111   133   221
2                      Asia   282   350   411   502   635   809   947  1402
3                    Europe   168   170   178   190   203   276   408   547
4     Latin America[Note 1]    40    20    10    16    24    38    74   167
5  Northern America[Note 1]     6     3     2     2     7    26    82   172
6                   Oceania     3     3     3     2     2     2     6    13

   1999  2008  2010  2012  2050  2150
0  6008  6707  6896  7052  9725  9746
1   783   973  1022  1052  2478  2308
2  3700  4054  4164  4250  5267  5561
3   675   732   738   740   734   517
4   508   577   590   603   784   912
5   312   337   345   351   433   398
6    30    34    37    38    57    51  ,
                     Region  1500  1600  1700  1750  1800  1850  1900  1950  \
0                    Africa  14.7  17.3  14.9  13.4  10.9   8.8   8.1   8.8
1                      Asia  48.2  53.0  57.9  63.5  64.9  64.1  57.4  55.6
2                    Europe  28.7  25.8  25.1  20.6  20.8  21.9  24.7  21.7
3     Latin America[Note 1]   6.8   3.0   1.4   2.0   2.5   3.0   4.5   6.6
4  Northern America[Note 1]   1.0   0.5   0.3   0.3   0.7   2.1   5.0   6.8
5                   Oceania   0.5   0.5   0.4   0.3   0.2   0.2   0.4   0.5

   1999  2008  2010  2012  2050  2150
0  13.0  14.5  14.8  15.2  25.5  23.7
1  61.6  60.4  60.4  60.3  54.2  57.1
2  11.2  10.9  10.7  10.5   7.6   5.3
3   8.5   8.6   8.6   8.6   8.1   9.4
4   5.2   5.0   5.0   5.0   4.5   4.1
5   0.5   0.5   0.5   0.5   0.6   0.5  ,
          Year    World  Africa    Asia  Europe  Latin America& Carib.[Note 1]  \
0    70,000 BC  < 0.015     NaN     NaN     NaN                             0.0
1    10,000 BC        4     NaN     NaN     NaN                             NaN
2      8000 BC        5     NaN     NaN     NaN                             NaN
3      6500 BC        5     NaN     NaN     NaN                             NaN
4      5000 BC        5     NaN     NaN     NaN                             NaN
5      4000 BC        7     NaN     NaN     NaN                             NaN
6      3000 BC       14     NaN     NaN     NaN                             NaN
7      2000 BC       27     NaN     NaN     NaN                             NaN
8      1000 BC       50     7.0    33.0     9.0                             NaN
9       500 BC      100    14.0    66.0    16.0                             NaN
10        AD 1      200    23.0   141.0    28.0                             NaN
11        1000      400    70.0   269.0    50.0                             8.0
12        1500      458    86.0   243.0    84.0                            39.0
13        1600      580   114.0   339.0   111.0                            10.0
14        1700      682   106.0   436.0   125.0                            10.0
15        1750      791   106.0   502.0   163.0                            16.0
16        1800     1000   107.0   656.0   203.0                            24.0
17        1850     1262   111.0   809.0   276.0                            38.0
18        1900     1650   133.0   947.0   408.0                            74.0
19        1950     2525   229.0  1394.0   549.0                           169.0
20        1955     2758   254.0  1534.0   577.0                           193.0
21        1960     3018   285.0  1687.0   606.0                           221.0
22        1965     3322   322.0  1875.0   635.0                           254.0
23        1970     3682   366.0  2120.0   657.0                           288.0
24        1975     4061   416.0  2378.0   677.0                           326.0
25        1980     4440   478.0  2626.0   694.0                           365.0
26        1985     4853   550.0  2897.0   708.0                           406.0
27        1990     5310   632.0  3202.0   721.0                           447.0
28        1995     5735   720.0  3475.0   728.0                           487.0
29        2000     6127   814.0  3714.0   726.0                           527.0
30        2005     6520   920.0  3945.0   729.0                           564.0
31        2010     6930  1044.0  4170.0   735.0                           600.0
32        2015     7349  1186.0  4393.0   738.0                           634.0
```

	North America[Note 1]	Oceania	Notes
0	0.0	NaN	[121]
1	NaN	NaN	[122]
2	NaN	NaN	NaN
3	NaN	NaN	NaN
4	NaN	NaN	NaN
5	NaN	NaN	NaN
6	NaN	NaN	NaN
7	NaN	NaN	NaN
8	NaN	NaN	[citation needed]
9	NaN	NaN	NaN
10	NaN	NaN	NaN
11	1.0	2.0	NaN
12	3.0	3.0	NaN
13	3.0	3.0	NaN
14	2.0	3.0	NaN
15	2.0	2.0	NaN
16	7.0	3.0	NaN
17	26.0	2.0	NaN
18	82.0	6.0	NaN
19	172.0	12.7	[123]
20	187.0	14.2	NaN
21	204.0	15.8	NaN
22	219.0	17.5	NaN
23	231.0	19.7	NaN
24	242.0	21.5	NaN
25	254.0	23.0	NaN
26	267.0	24.9	NaN
27	281.0	27.0	NaN
28	296.0	29.1	NaN
29	314.0	31.1	NaN
30	329.0	33.4	NaN
31	344.0	36.4	NaN
32	358.0	39.3	NaN

,

	0	1
0	NaN	This section needs additional citations for ve...,

	Year	UN est.(millions)	Difference	USCB est.(millions)	Difference.1
0	2005	6542	-	6473	-
1	2010	6957	415	6866	393
2	2015	7380	423	7256	390
3	2020	7795	415	7643	380
4	2025	8184	390	8007	363
5	2030	8549	364	8341	334
6	2035	8888	339	8646	306
7	2040	9199	311	8926	280
8	2045	9482	283	9180	254
9	2050	9735	253	9408	228,

	Year	World	Asia	Africa	Europe \
0	2000	6144	3,741 (60.9%)	811 (13.2%)	726 (11.8%)
1	2005	6542	3,978 (60.8%)	916 (14.0%)	729 (11.2%)
2	2010	6957	4,210 (60.5%)	1,039 (14.9%)	736 (10.6%)
3	2015	7380	4,434 (60.1%)	1,182 (16.0%)	743 (10.1%)
4	2020	7795	4,641 (59.5%)	1,341 (17.2%)	748 (9.6%)
5	2025	8184	4,823 (58.9%)	1,509 (18.4%)	746 (9.1%)
6	2030	8549	4,974 (58.2%)	1,688 (19.8%)	741 (8.7%)
7	2035	8888	5,096 (57.3%)	1,878 (21.1%)	735 (8.3%)
8	2040	9199	5,189 (56.4%)	2,077 (22.6%)	728 (7.9%)
9	2045	9482	5,253 (55.4%)	2,282 (24.1%)	720 (7.6%)
10	2050	9735	5,290 (54.3%)	2,489 (25.6%)	711 (7.3%)
11	2055	9958	5,302 (53.2%)	2,698 (27.1%)	700 (7.0%)
12	2060	10152	5,289 (52.1%)	2,905 (28.6%)	689 (6.8%)
13	2065	10318	5,256 (51.0%)	3,109 (30.1%)	677 (6.6%)
14	2070	10459	5,207 (49.8%)	3,308 (31.6%)	667 (6.4%)
15	2075	10577	5,143 (48.6%)	3,499 (33.1%)	657 (6.2%)
16	2080	10674	5,068 (47.5%)	3,681 (34.5%)	650 (6.1%)
17	2085	10750	4,987 (46.4%)	3,851 (35.8%)	643 (6.0%)
18	2090	10810	4,901 (45.3%)	4,008 (37.1%)	638 (5.9%)
19	2095	10852	4,812 (44.3%)	4,152 (38.3%)	634 (5.8%)
20	2100	10875	4,719 (43.4%)	4,280 (39.4%)	630 (5.8%)

	Latin America/Caribbean	Northern America	Oceania

```
 0               522 (8.5%)        312 (5.1%)  31 (0.5%)
 1               558 (8.5%)        327 (5.0%)  34 (0.5%)
 2               591 (8.5%)        343 (4.9%)  37 (0.5%)
 3               624 (8.5%)        357 (4.8%)  40 (0.5%)
 4               654 (8.4%)        369 (4.7%)  43 (0.6%)
 5               682 (8.3%)        380 (4.6%)  45 (0.6%)
 6               706 (8.3%)        391 (4.6%)  48 (0.6%)
 7               726 (8.2%)        401 (4.5%)  50 (0.6%)
 8               742 (8.1%)        410 (4.5%)  53 (0.6%)
 9               754 (8.0%)        418 (4.4%)  55 (0.6%)
 10              762 (7.8%)        425 (4.4%)  57 (0.6%)
 11              767 (7.7%)        432 (4.3%)  60 (0.6%)
 12              768 (7.6%)        439 (4.3%)  62 (0.6%)
 13              765 (7.4%)        447 (4.3%)  64 (0.6%)
 14              759 (7.3%)        454 (4.3%)  66 (0.6%)
 15              750 (7.1%)        461 (4.4%)  67 (0.6%)
 16              739 (6.9%)        468 (4.4%)  69 (0.7%)
 17              726 (6.8%)        474 (4.4%)  71 (0.7%)
 18              711 (6.6%)        479 (4.4%)  72 (0.7%)
 19              696 (6.4%)        485 (4.5%)  74 (0.7%)
 20              680 (6.3%)        491 (4.5%)  75 (0.7%)  ,
   Population(in billions)   0.5  0.5.1     1   1.1     2   2.1     4   4.1  \
 0                    Year  1500   1500  1804  1804  1927  1927  1974  1974
 1           Years elapsed     —    304   304   123   123    47    47    48

      8   8.1                                                 16  \
 0  2022  2022  .mw-parser-output .tooltip-dotted{border-botto...
 1    48    48                                                 —

                                          16.1
 0  .mw-parser-output .tooltip-dotted{border-botto...
 1                                         NaN  ,
   Population(in billions) 0.375  0.375.1  0.75  0.75.1   1.5  1.5.1     3  \
 0                    Year  1171     1171  1715    1715  1881   1881  1960
 1           Years elapsed     —      544   544     166   166     79    79

    3.1     6   6.1           12          12.1
 0  1960  1999  1999  c. 2100[146]  c. 2100[146]
 1    39    39    39       c. 100+       c. 100+  ,
   .mw-parser-output .navbar{display:inline;font-size:88%;font-weight:normal}.mw-parser-o
utput .navbar-collapse{float:left;text-align:left}.mw-parser-output .navbar-boxtext{word-
spacing:0}.mw-parser-output .navbar ul{display:inline-block;white-space:nowrap;line-heigh
t:inherit}.mw-parser-output .navbar-brackets::before{margin-right:-0.125em;content:"[ "}.
mw-parser-output .navbar-brackets::after{margin-left:-0.125em;content:" ]"}.mw-parser-out
put .navbar li{word-spacing:-0.125em}.mw-parser-output .navbar a>span,.mw-parser-output .
navbar a>abbr{text-decoration:inherit}.mw-parser-output .navbar-mini abbr{font-variant:sm
all-caps;border-bottom:none;text-decoration:none;cursor:inherit}.mw-parser-output .navbar
-ct-full{font-size:114%;margin:0 7em}.mw-parser-output .navbar-ct-mini{font-size:114%;mar
gin:0 4em}vteGlobal human population  \
 0                                       Major topics

 1                         Biological andrelated topics

 2                                   Populationecology

 3                              Society and population

 4                                          Literature

 5                                        Publications

 6                                               Lists

 7                            Events andorganizations

 8                                      Related topics

 9                          Commons  Human overpopulation


   .mw-parser-output .navbar{display:inline;font-size:88%;font-weight:normal}.mw-parser-o
utput .navbar-collapse{float:left;text-align:left}.mw-parser-output .navbar-boxtext{word-
```

```
spacing:0}.mw-parser-output .navbar ul{display:inline-block;white-space:nowrap;line-heigh
t:inherit}.mw-parser-output .navbar-brackets::before{margin-right:-0.125em;content:"[ "}.
mw-parser-output .navbar-brackets::after{margin-left:-0.125em;content:" ]"}.mw-parser-out
put .navbar li{word-spacing:-0.125em}.mw-parser-output .navbar a>span,.mw-parser-output .
navbar a>abbr{text-decoration:inherit}.mw-parser-output .navbar-mini abbr{font-variant:sm
all-caps;border-bottom:none;text-decoration:none;cursor:inherit}.mw-parser-output .navbar
-ct-full{font-size:114%;margin:0 7em}.mw-parser-output .navbar-ct-mini{font-size:114%;mar
gin:0 4em}vteGlobal human population.1
 0  Demographics of the world Demographic transiti...

 1  Population biology Population decline Populati...

 2  Earth's energy budget I = P × A × T Kaya ident...

 3  Biocapacity Human overpopulation Malthusian ca...

 4  A Modest Proposal Observations Concerning the ...

 5  Population and Environment Population and Deve...

 6  Population and housing censuses by country Lar...

 7  7 Billion Actions International Conference on ...

 8  Deep ecology Fertility and intelligence Green ...

 9                      Commons  Human overpopulation
,
            Articles related to the world's population  \
 0   vteHuman impact on the environmentGeneral Anth...
 1                 vteHuman impact on the environment
 2                                            General
 3                                             Causes
 4                                            Effects
 5                                         Mitigation
 6   Commons  Category by country assessment mitiga...
 7      vteLists of countries by population statistics
 8                                             Global
 9                               Continents/subregions
 10                                   Intercontinental
 11                                 Cities/urban areas
 12                                    Past and future
 13                                 Population density
 14                                  Growth indicators
 15                                 Other demographics
 16                                             Health
 17                             Education and innovation
 18                                           Economic
 19    List of international rankings Lists by country
 20                                 vteHierarchy of life
 21  Biosphere > Biome > Ecosystem > Biocenosis > P...
 22                                   vteGlobalization
 23                         Journals  Outline Studies
 24                                            Aspects
 25                                             Issues
 26                                             Global
 27                                              Other
 28                                           Theories
 29                                  Notablescholars
 30                                          Economics
 31                                  Political economy
 32                               Politics / sociology
 33                                       Non-academic
 34                          Category  Business portal

          Articles related to the world's population.1
 0   vteHuman impact on the environmentGeneral Anth...
 1                 vteHuman impact on the environment
 2   Anthropocene Environmental issues list of issu...
 3   Agriculture cannabis cultivation irrigation me...
 4   Biodiversity threats biodiversity loss decline...
 5   Alternative fuel vehicle propulsion Birth cont...
```

```
6   Commons  Category by country assessment mitiga...
7      vteLists of countries by population statistics
8   Current population United Nations Demographics...
9   Africa Antarctica Asia Europe North America Ca...
10  Americas Arab world Commonwealth of Nations Eu...
11  World cities National capitals Megacities Mega...
12  Past and future population World population es...
13  Current density Past and future population den...
14  Population growth rate Natural increase Net re...
15  Age at childbearing Age at first marriage Age ...
16  Antidepressant consumption Antiviral medicatio...
17  Bloomberg Innovation Index Education Index Int...
18  Access to financial services Development aid d...
19    List of international rankings Lists by country
20                              vteHierarchy of life
21  Biosphere > Biome > Ecosystem > Biocenosis > P...
22                                  vteGlobalization
23                         Journals  Outline Studies
24  Alter-globalization Anti-globalization Cultura...
25  Global Climate change Climate justice Disease ...
26  Climate change Climate justice Disease COVID-1...
27  Brain drain reverse Care drain Development aid...
28  Capital accumulation Dependency Development Ea...
29  Economics David Autor Richard Baldwin Ravi Bat...
30  David Autor Richard Baldwin Ravi Batra Jagdish...
31  Samir Amin Giovanni Arrighi Robert W. Cox Andr...
32  Arjun Appadurai Daniele Archibugi K. Anthony A...
33  Noam Chomsky Thomas Friedman Naomi Klein John ...
34                         Category  Business portal  ,
                 vteHuman impact on the environment  \
0                                            General
1                                             Causes
2                                            Effects
3                                         Mitigation
4  Commons  Category by country assessment mitiga...

               vteHuman impact on the environment.1
0  Anthropocene Environmental issues list of issu...
1  Agriculture cannabis cultivation irrigation me...
2  Biodiversity threats biodiversity loss decline...
3  Alternative fuel vehicle propulsion Birth cont...
4  Commons  Category by country assessment mitiga...  ,
      vteLists of countries by population statistics  \
0                                             Global
1                              Continents/subregions
2                                   Intercontinental
3                                 Cities/urban areas
4                                    Past and future
5                                 Population density
6                                  Growth indicators
7                                 Other demographics
8                                             Health
9                            Education and innovation
10                                          Economic
11  List of international rankings Lists by country

      vteLists of countries by population statistics.1
0   Current population United Nations Demographics...
1   Africa Antarctica Asia Europe North America Ca...
2   Americas Arab world Commonwealth of Nations Eu...
3   World cities National capitals Megacities Mega...
4   Past and future population World population es...
5   Current density Past and future population den...
6   Population growth rate Natural increase Net re...
7   Age at childbearing Age at first marriage Age ...
8   Antidepressant consumption Antiviral medicatio...
9   Bloomberg Innovation Index Education Index Int...
10  Access to financial services Development aid d...
11    List of international rankings Lists by country  ,
                             vteHierarchy of life  \
0  Biosphere > Biome > Ecosystem > Biocenosis > P...
```

```
                                                vteHierarchy of life.1
0  Biosphere > Biome > Ecosystem > Biocenosis > P...  ,
                      vteGlobalization  \
0   Journals  Outline Studies
1                     Aspects
2                      Issues
3                      Global
4                       Other
5                    Theories
6              Notablescholars
7                   Economics
8           Political economy
9        Politics / sociology
10               Non-academic
11  Category  Business portal

                                             vteGlobalization.1
0                              Journals  Outline Studies
1   Alter-globalization Anti-globalization Cultura...
2   Global Climate change Climate justice Disease ...
3   Climate change Climate justice Disease COVID-1...
4   Brain drain reverse Care drain Development aid...
5   Capital accumulation Dependency Development Ea...
6   Economics David Autor Richard Baldwin Ravi Bat...
7   David Autor Richard Baldwin Ravi Batra Jagdish...
8   Samir Amin Giovanni Arrighi Robert W. Cox Andr...
9   Arjun Appadurai Daniele Archibugi K. Anthony A...
10  Noam Chomsky Thomas Friedman Naomi Klein John ...
11                             Category  Business portal  ,
        0                                                   1
0  Global  Climate change Climate justice Disease COVID-1...
1   Other  Brain drain reverse Care drain Development aid...,
                      0                                                   1
0             Economics  David Autor Richard Baldwin Ravi Batra Jagdish...
1     Political economy  Samir Amin Giovanni Arrighi Robert W. Cox Andr...
2  Politics / sociology  Arjun Appadurai Daniele Archibugi K. Anthony A...
3          Non-academic  Noam Chomsky Thomas Friedman Naomi Klein John ...,
                                         0        1
0  Authority control: National libraries  Germany]
```

Not Useful Tables

Pandas found 24 tables on that page. Some are not useful:

In [8]:

```
len(tables) #Here total number of tables
```

Out[8]:

24

Tables that need formatting

Some will be misaligned, meaning you need to do extra work to fix the columns and rows:

In [9]:

```
tables[0] # Here first table from page
```

Out[9]:

	Population	1	2	3	4	5	6	7	8	9	10
0	Year	1804	1930	1960	1974	1987	1999	2011	2022	2037	2057
1	Years elapsed	200,000+	126	30	14	13	12	12	11	15	20

In [18]:

```
world_topten = tables[1]
world_topten
```

Out[18]:

	#	Most populous countries	2000	2015	2030[A]
0	1	China[B]	1270	1376	1416
1	2	India	1053	1311	1528
2	3	United States	283	322	356
3	4	Indonesia	212	258	295
4	5	Pakistan	136	208	245
5	6	Brazil	176	206	228
6	7	Nigeria	123	182	263
7	8	Bangladesh	131	161	186
8	9	Russia	146	146	149
9	10	Mexico	103	127	148
10	NaN	World total	6127	7349	8501
11	Notes: .mw-parser-output .reflist{font-size:90...	Notes: .mw-parser-output .reflist{font-size:90...	Notes: .mw-parser-output .reflist{font-size:90...	Notes: .mw-parser-output .reflist{font-size:90...	Notes: .mw-parser-output .reflist{font-size:90...

In [19]:

```
#Here we are going to remove 11 row
world_topten = world_topten.drop(11,axis=0)
world_topten
```

Out[19]:

	#	Most populous countries	2000	2015	2030[A]
0	1	China[B]	1270	1376	1416
1	2	India	1053	1311	1528
2	3	United States	283	322	356
3	4	Indonesia	212	258	295
4	5	Pakistan	136	208	245
5	6	Brazil	176	206	228
6	7	Nigeria	123	182	263
7	8	Bangladesh	131	161	186
8	9	Russia	146	146	149
9	10	Mexico	103	127	148
10	NaN	World total	6127	7349	8501

In [20]:

```
# Here we are going to drop # column
world_topten = world_topten.drop('#',axis=1)
world_topten
```

Out[20]:

	Most populous countries	2000	2015	2030[A]
0	China[B]	1270	1376	1416
1	India	1053	1311	1528

	Most populous countries	2000	2015	2030[A]
2	United States	283	322	356
3	Indonesia	212	258	295
4	Pakistan	136	208	245
5	Brazil	176	206	228
6	Nigeria	123	182	263
7	Bangladesh	131	161	186
8	Russia	146	146	149
9	Mexico	103	127	148
10	World total	6127	7349	8501

In [22]:

```
# Renaming columns by own
world_topten.columns = ['Country','2000','2015','2030 Est.']
world_topten
```

Out[22]:

	Country	2000	2015	2030 Est.
0	China[B]	1270	1376	1416
1	India	1053	1311	1528
2	United States	283	322	356
3	Indonesia	212	258	295
4	Pakistan	136	208	245
5	Brazil	176	206	228
6	Nigeria	123	182	263
7	Bangladesh	131	161	186
8	Russia	146	146	149
9	Mexico	103	127	148
10	World total	6127	7349	8501

In [23]:

```
tables[6]
```

Out[23]:

	Year	Population	Yearly growth		Density(pop/km2)	Urban population	
	Year	Population	%	Number	Density(pop/km2)	Number	%
0	1951	2584034261	1.88%	47603112	17	775067697	30%
1	1952	2630861562	1.81%	46827301	18	799282533	30%
2	1953	2677608960	1.78%	46747398	18	824289989	31%
3	1954	2724846741	1.76%	47237781	18	850179106	31%
4	1955	2773019936	1.77%	48173195	19	877008842	32%
...	...	...	...	...	...	...	...
65	2016	7464022000	1.14%	84225000	50	4060653000	54%
66	2017	7547859000	1.12%	83837000	51	4140189000	55%
67	2018	7631091000	1.10%	83232000	51	4219817000	55%
68	2019	7713468000	1.08%	82377000	52	4299439000	56%
69	2020	7795000000	1.05%	81331000	52	4378900000	56%

70 rows × 7 columns

Write to html Output

If you are working on a website and want to quickly output the .html file, you can use to_html

In [41]:

```
# Saving table this to html
world_topten.to_html('D:\Study\sample.html',index= False)
```

read_html is not perfect, but its quite powerful for such a simple method call!

In [ ]:

```

```

Excel Files

Pandas can read in basic excel files (it will get errors if there are macros or extensive formulas relying on outside excel files), in general, pandas can only grab the raw information from an .excel file.

NOTE: Requires the openpyxl and xlrd library! Its provided for you in our environment, or simply install with:

```
pip install openpyxl
pip install xlrd
```

Heavy excel users may want to check out this website: https://www.python-excel.org/

You can think of an excel file as a Workbook containin sheets, which for pandas means each sheet can be a DataFrame.

Excel file input with read_excel()

In [47]:

```
df = pd.read_excel("D:\\Study\\Programming\\python\\Python course from udemy\\[GigaCourse.Com] Udemy - 2022 Python for Machine Learning & Data Science Masterclass\\01 - Introduction to Course\\1UNZIP-FOR-NOTEBOOKS-FINAL\\03-Pandas\\my_excel_file.xlsx"
                   ,sheet_name='First_Sheet')
df # Here we can give sheet name if there are more than one sheets , for single sheet we can skip that
```

Out[47]:

	a	b	c	d
0	0	1	2	3
1	4	5	6	7
2	8	9	10	11
3	12	13	14	15

What if you don't know the sheet name? Or want to run a for loop for certain sheet names? Or want every sheet?

Several ways to do this: https://stackoverflow.com/questions/17977540/pandas-looking-up-the-list-of-sheets-in-an-excel-file

In [50]:

```
# Suppose there were many sheet in excel file and we want name of all sheet so we can try that
wb = pd.ExcelFile("D:\\Study\\Programming\\python\\Python course from udemy\\[GigaCourse.Com] Udemy - 2022 Python for Machine Learning & Data Science Masterclass\\01 - Introduction to Course\\1UNZIP-FOR-NOTEBOOKS-FINAL\\03-Pandas\\my_excel_file.xlsx")
wb.sheet_names
```

Out[50]:

```
['First_Sheet']
```

In [55]:

```
# or  see end
pd.ExcelFile("D:\\Study\\Programming\\python\\Python course from udemy\\[GigaCourse.Com] Udemy - 2022 Python for Machine Learning & Data Science Masterclass\\01 - Introduction to Course\\1UNZIP-FOR-NOTEBOOKS-FINAL\\03-Pandas\\my_excel_file.xlsx").sheet_names
```

Out[55]:

```
['First_Sheet']
```

Grab all sheets

In [58]:

```
excel_sheets = pd.read_excel("D:\\Study\\Programming\\python\\Python course from udemy\\[GigaCourse.Com] Udemy - 2022 Python for Machine Learning & Data Science Masterclass\\01 - Introduction to Course\\1UNZIP-FOR-NOTEBOOKS-FINAL\\03-Pandas\\my_excel_file.xlsx"
                              , sheet_name=None)
```

In [59]:

```
type(excel_sheets) # it has store all sheet name as in form of dict
```

Out[59]:

```
dict
```

In [61]:

```
excel_sheets.keys() # Here we can call those sheet name
```

Out[61]:

```
dict_keys(['First_Sheet'])
```

In [63]:

```
df=excel_sheets['First_Sheet']
df
```

Out[63]:

	a	b	c	d
0	0	1	2	3
1	4	5	6	7
2	8	9	10	11
3	12	13	14	15

Saving Excel file

In [65]:

```
df.to_excel('D:\\Study\\sample_excel.xlsx',sheet_name= 'First_Sheet', index = False)
```

```
df.to_excel('D:\\Study\\sample_excel.xlsx',sheet_name='First_Sheet',index=False)
```

SQL Connections

NOTE: Highly recommend you explore specific libraries for your specific SQL Engine. Simple search for your database+python in Google and the top results should hopefully include an API.

- [MySQL]
- [PostgreSQL]
- [MS SQL Server]
- [Orcale]
- [MongoDB]

Let's review pandas capabilities by using SQLite, which comes built in with Python.

Example SQL Database (temporary in your RAM)

You will need to install sqlalchemy with:

```
pip install sqlalchemy
```

to follow along. To understand how to make a connection to your own database, make sure to review: https://docs.sqlalchemy.org/en/13/core/connections.html

In [66]:

```
from sqlalchemy import create_engine
```

In [70]:

```
temp_db = create_engine('sqlite:///:memory:')
```

Write to Database

In [76]:

```
tables[5]
```

Out[76]:

	Rank	Country	Population	Area(km2)	Density(pop/km2)	Population trend[citation needed]
0	1	India	1389637446	3287263	423	Growing
1	2	Pakistan	242923845	796095	305	Rapidly growing
2	3	Bangladesh	165650475	148460	1116	Rapidly growing
3	4	Japan	124214766	377915	329	Declining[104]
4	5	Philippines	114597229	300000	382	Growing
5	6	Vietnam	103808319	331210	313	Growing
6	7	United Kingdom	67791400	243610	278	Growing
7	8	South Korea	51844834	99720	520	Steady
8	9	Taiwan	23580712	35980	655	Steady
9	10	Sri Lanka	23187516	65610	353	Growing

In [80]:

```
pop = tables[5]
```

In [83]:

```
pop.to_sql(name='populations1',con=temp_db)
```

Out[83]:

10

Read from SQL Database

In [84]:

```
# Read in an entire table
pd.read_sql(sql='populations1',con=temp_db)
```

Out[84]:

	index	Rank	Country	Population	Area(km2)	Density(pop/km2)	Population trend[citation needed]
0	0	1	India	1389637446	3287263	423	Growing
1	1	2	Pakistan	242923845	796095	305	Rapidly growing
2	2	3	Bangladesh	165650475	148460	1116	Rapidly growing
3	3	4	Japan	124214766	377915	329	Declining[104]
4	4	5	Philippines	114597229	300000	382	Growing
5	5	6	Vietnam	103808319	331210	313	Growing
6	6	7	United Kingdom	67791400	243610	278	Growing
7	7	8	South Korea	51844834	99720	520	Steady
8	8	9	Taiwan	23580712	35980	655	Steady
9	9	10	Sri Lanka	23187516	65610	353	Growing

In [85]:

```
# Read in with a SQL Query
pd.read_sql_query(sql="SELECT Country FROM populations1",con=temp_db)
```

Out[85]:

	Country
0	India
1	Pakistan
2	Bangladesh
3	Japan
4	Philippines
5	Vietnam
6	United Kingdom
7	South Korea
8	Taiwan
9	Sri Lanka

It is difficult to generalize pandas and SQL, due to a wide array of issues, including permissions,security, online access, varying SQL engines, etc... Use these ideas as a starting off point, and you will most likely need to do your own research for your own situation.

In [ ]:

```

```