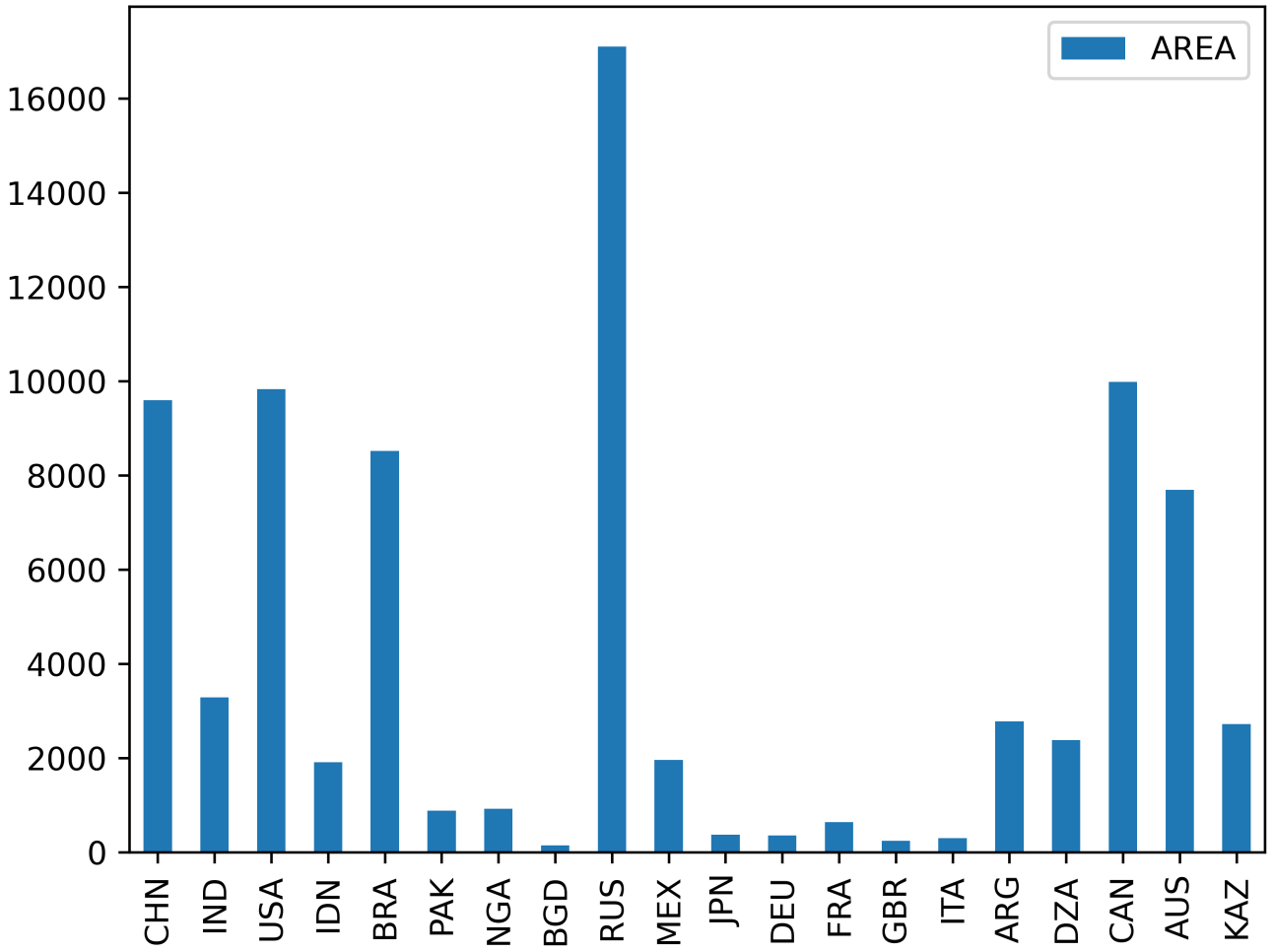
PANDAS: HOW TO READ AND WRITE FILES

Pandas: How to Read and Write Files

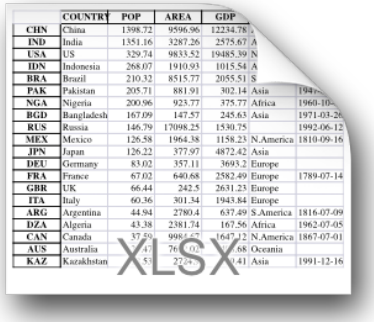
Pandas: How to Read and Write Files

```
>>> df.describe()
                POP          AREA           GDP
count     20.000000     20.000000     20.00000
mean     248.905500   4082.182500   3036.14250
std      394.546143   4706.507539   4706.00783
min       18.530000    147.570000    159.41000
25%       56.505000    575.002500    572.06000
50%      126.400000   2173.060000   1588.93500
75%      206.862500   7897.957500   2594.67500
max     1398.720000  17098.250000  19485.39000
```

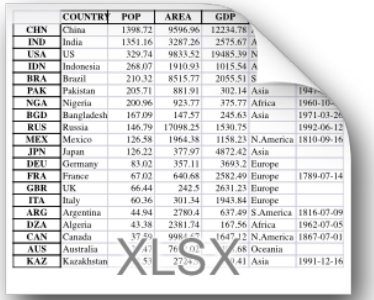
Pandas: How to Read and Write Files

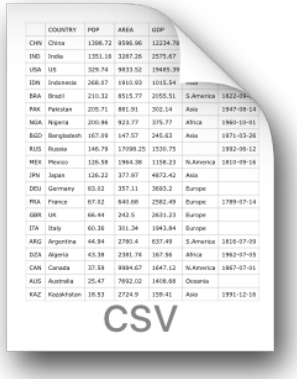


Pandas: How to Read and Write Files

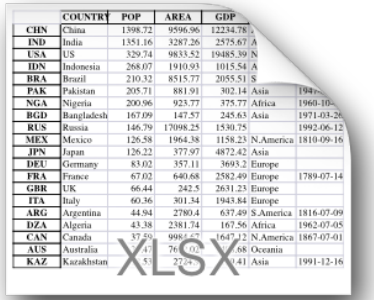


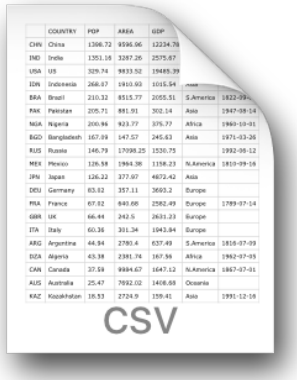
Pandas: How to Read and Write Files



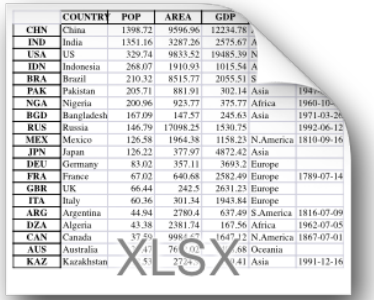


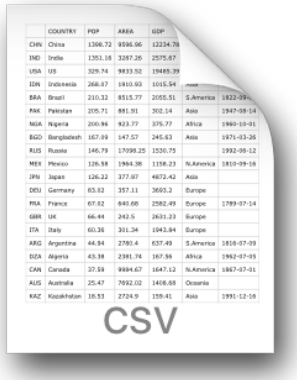
Pandas: How to Read and Write Files



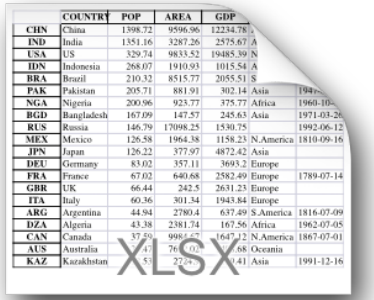


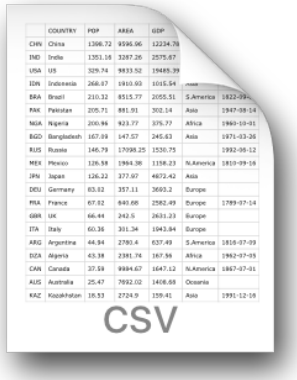
Pandas: How to Read and Write Files



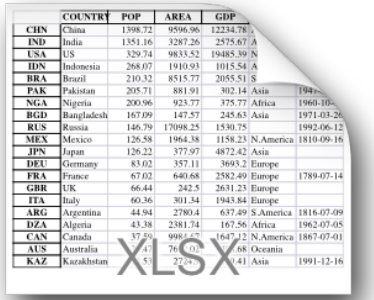


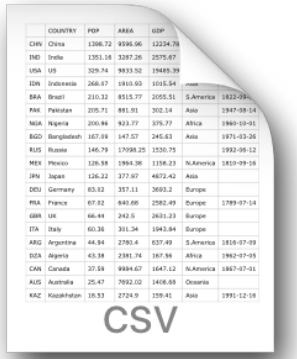
Pandas: How to Read and Write Files





Pandas: How to Read and Write Files





Pandas: How to Read and Write Files

Pandas: How to Read and Write Files

- Read and Write Data
- The Pandas IOTools API
- Working With Different File Formats
- Working With Big Data

Let’s Get Started!

Pandas: How to Read and Write Files

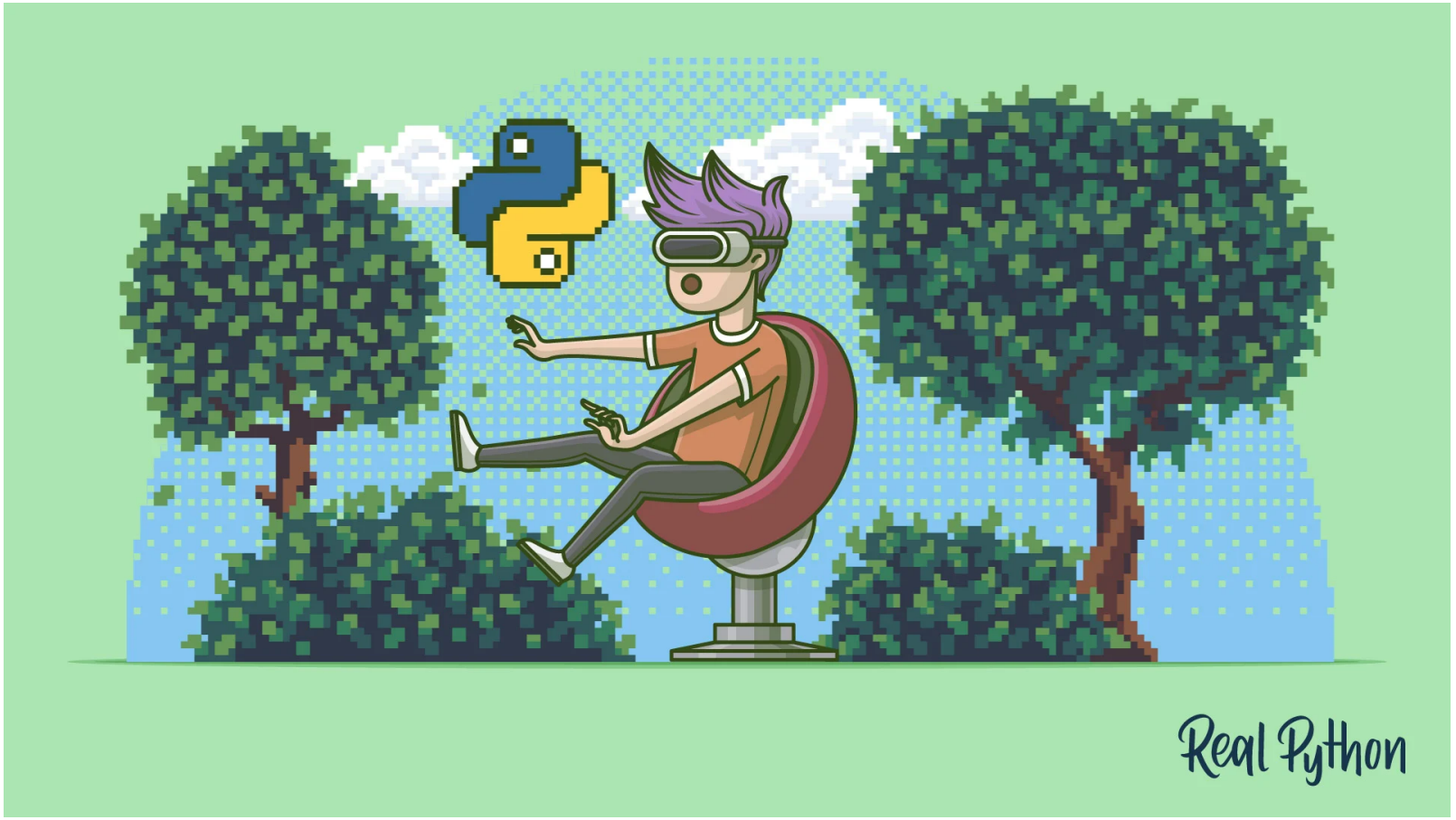
Pandas: How to Read and Write Files

1. How To Read and Write Files
   - 1.1 **Installing Pandas and Preparing Data**
   - 1.2 Reading and Writing CSV Files
   - 1.3 Reading and Writing Excel Files
2. Working With Different File Types
3. Working With Big Data

Installing Pandas

- Python 3.9
- `Pandas==1.2.5`

Python Virtual Environments Primer



https://realpython.com/python-virtual-environments-a-primer/

Preparing Data

- Country - `COUNTRY`
- Population - `POP`
- Area - `AREA`
- Gross Domestic Product - `GDP`
- Continent - `CONT`
- Independence Day - `IND_DAY`

Code	Country	Pop	Area	GDP	CONT	IND_DAY
CHN	China	1398.72	9596.96	12234.8	Asia	
IND	India	1351.16	3287.26	2575.67	Asia	1947-08-15
USA	US	329.74	9833.52	19485.4	N.America	1776-07-04
IDN	Indonesia	268.07	1910.93	1015.54	Asia	1945-08-17
BRA	Brazil	210.32	8515.77	2055.51	S.America	1822-09-07
PAK	Pakistan	205.71	881.91	302.14	Asia	1947-08-14
NGA	Nigeria	200.96	923.77	375.77	Africa	1960-10-01
BGD	Bangladesh	167.09	147.57	245.63	Asia	1971-03-26
RUS	Russia	146.79	17098.2	1530.75		
MEX	Mexico	126.58	1964.38	1158.23	N.America	1810-09-16
JPN	Japan	126.22	377.97	4872.42	Asia	
DEU	Germany	83.02	357.11	3693.2	Europe	
FRA	France	67.02	640.68	2582.49	Europe	1789-07-14
GBR	UK	66.44	242.5	2631.23	Europe	
ITA	Italy	60.36	301.34	1943.84	Europe	
ARG	Argentina	44.94	2780.4	637.49	S.America	1816-07-09
DZA	Algeria	43.38	2381.74	167.56	Africa	1962-07-05
CAN	Canada	37.59	9984.67	1647.12	N.America	1867-07-01
AUS	Australia	25.47	7692.02	1408.68	Oceania	
KAZ	Kazakhstan	18.53	2724.9	159.41	Asia	1991-12-16

`data.py` - Dictionary Structure

```
data = {
    'CHN': {'COUNTRY': 'China', 'POP': 1_398.72, 'AREA': 9_596.96,
            'GDP': 12_234.78, 'CONT': 'Asia'},
}
```

`data.py` - Dictionary Structure

```
data = {
    'CHN': {'COUNTRY': 'China', 'POP': 1_398.72, 'AREA': 9_596.96,
            'GDP': 12_234.78, 'CONT': 'Asia'},
    'IND': {'COUNTRY': 'India', 'POP': 1_351.16, 'AREA': 3_287.26,
            'GDP': 2_575.67, 'CONT': 'Asia', 'IND_DAY': '1947-08-15'},
}
```

`data.py` - Dictionary Structure

```
data = {
    'CHN': {'COUNTRY': 'China', 'POP': 1_398.72, 'AREA': 9_596.96,
            'GDP': 12_234.78, 'CONT': 'Asia'},
    'IND': {'COUNTRY': 'India', 'POP': 1_351.16, 'AREA': 3_287.26,
            'GDP': 2_575.67, 'CONT': 'Asia', 'IND_DAY': '1947-08-15'},
    'USA': {'COUNTRY': 'US', 'POP': 329.74, 'AREA': 9_833.52,
            'GDP': 19_485.39, 'CONT': 'N.America',
            'IND_DAY': '1776-07-04'},
}
```

Next: Reading and Writing CSV Files

Pandas: How to Read and Write Files

1. How To Read and Write Files
   - 1.1 Installing Pandas and Preparing data
   - 1.2 **Reading and Writing CSV Files**
   - 1.3 Reading and Writing Excel Files
2. Working With Different File Types
3. Working With Big Data

CSV Files

- Plaintext File
- `.csv` Extension
- Holds Tabular Data
- Popular Format
- Each File Row Contains a Table Row
- Default Separator `,` can be Substituted

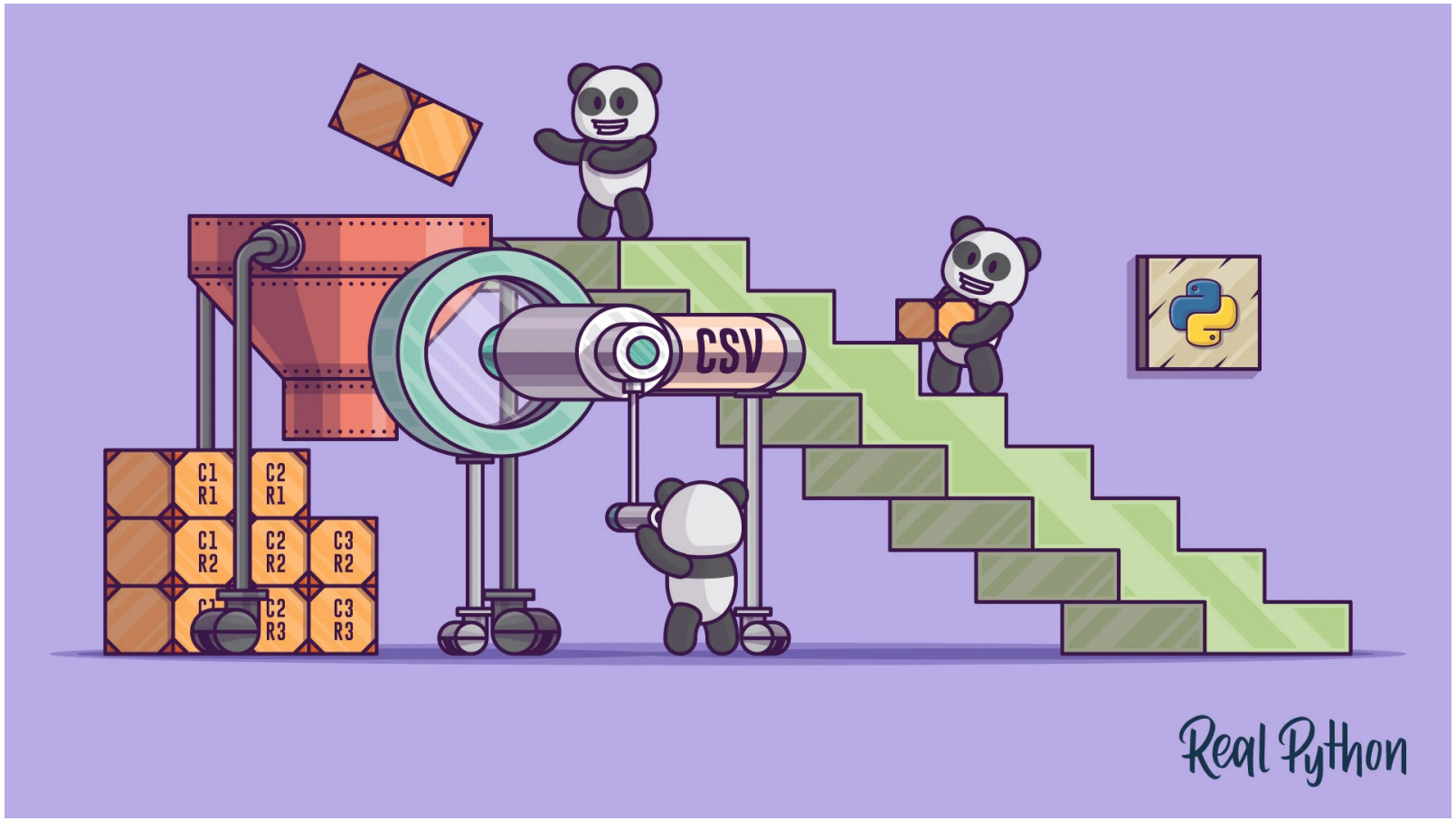
data.csv

```
,COUNTRY,POP,AREA,GDP,CONT,IND_DAY
CHN,China,1398.72,9596.96,12234.78,Asia,
IND,India,1351.16,3287.26,2575.67,Asia,1947-08-15
USA,US,329.74,9833.52,19485.39,N.America,1776-07-04
IDN,Indonesia,268.07,1910.93,1015.54,Asia,1945-08-17
BRA,Brazil,210.32,8515.77,2055.51,S.America,1822-09-07
PAK,Pakistan,205.71,881.91,302.14,Asia,1947-08-14
NGA,Nigeria,200.96,923.77,375.77,Africa,1960-10-01
BGD,Bangladesh,167.09,147.57,245.63,Asia,1971-03-26
RUS,Russia,146.79,17098.25,1530.75,,1992-06-12
MEX,Mexico,126.58,1964.38,1158.23,N.America,1810-09-16
JPN,Japan,126.22,377.97,4872.42,Asia,
DEU,Germany,83.02,357.11,3693.2,Europe,
FRA,France,67.02,640.68,2582.49,Europe,1789-07-14
GBR,UK,66.44,242.5,2631.23,Europe,
ITA,Italy,60.36,301.34,1943.84,Europe,
ARG,Argentina,44.94,2780.4,637.49,S.America,1816-07-09
DZA,Algeria,43.38,2381.74,167.56,Africa,1962-07-05
CAN,Canada,37.59,9984.67,1647.12,N.America,1867-07-01
AUS,Australia,25.47,7692.02,1408.68,Oceania,
KAZ,Kazakhstan,18.53,2724.9,159.41,Asia,1991-12-16
```

`index_col` specifies row labels

- Uses Zero-based Column Index
- Set when Rows Contain Labels
- Avoids Loading Labels as Data

Reading and Writing CSV Files in Python



https://realpython.com/python-csv/

Next: Reading and Writing Excel Files

Pandas: How to Read and Write Files

1. How To Read and Write Files
   - 1.1 Installing Pandas and Preparing Data
   - 1.2 Reading and Writing CSV Files
   - 1.3 **Reading and Writing Excel Files**
2. Working With Different File Types
3. Working With Big Data

Excel

- The Most Widely-Used Spreadsheet Software
- `.xls` - Binary Files
- `.xlsx` - XML-based Files
- Pandas Can Read and Write Excel Files
- Dependencies Must Be Installed for Some Formats

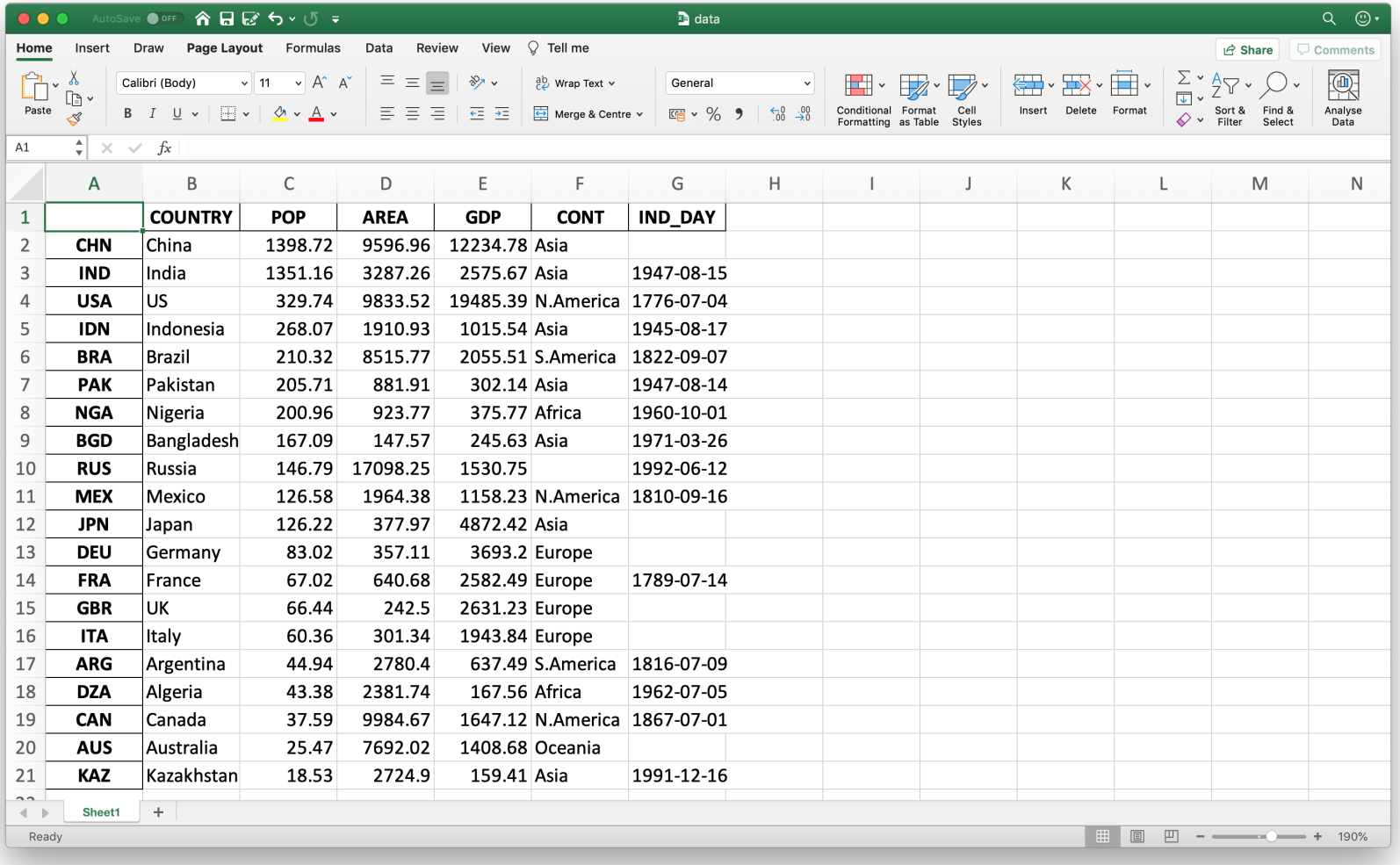
Pandas Excel Dependencies

- `xlwt` for `.xls` files
- `openpyxl` or `XlsxWriter` for `.xlsx` files
- `xlrd` to read Excel Files

Python Excel Library Capabilities

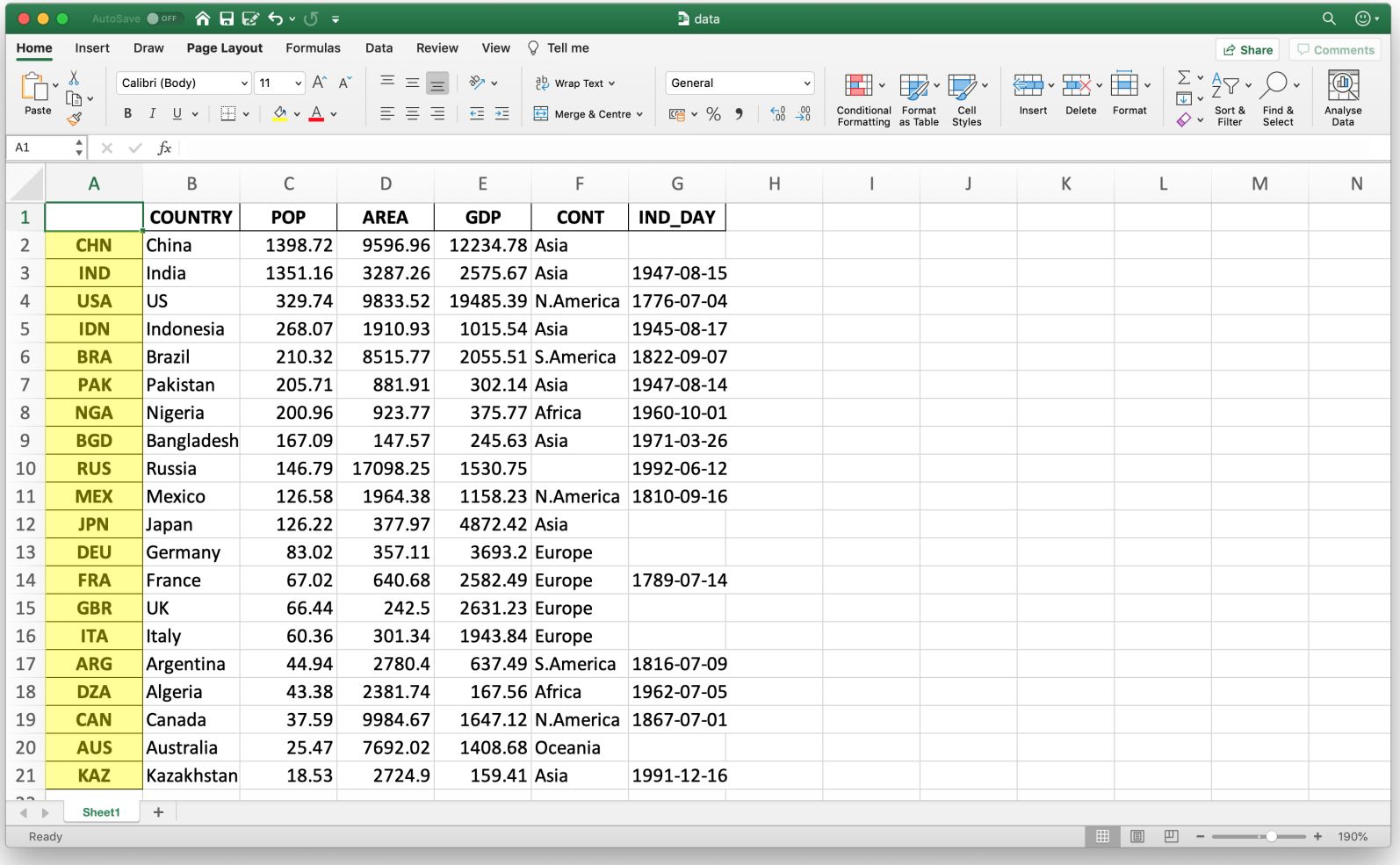
Library	File Type	Read	Write
`xlwt`	`.xls`	✗	✓
`xlrd`	`.xls`	✓	✗
`openpyxl`	`.xlsx`	✓	✓
`XlsxWriter`	`.xlsx`	✗	✓

`data.xlsx` viewed in Excel



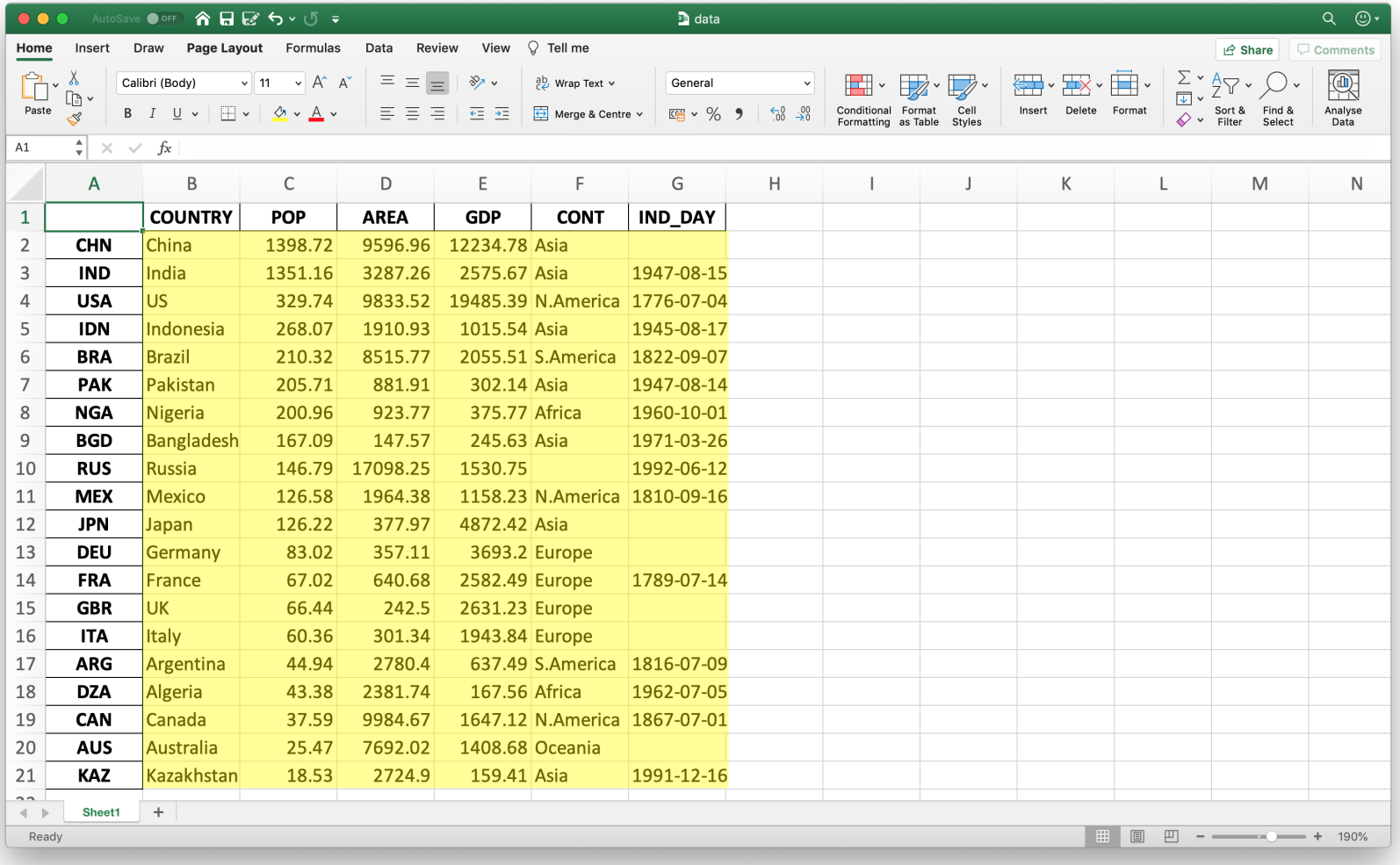
	A	B	C	D	E	F	G
1		COUNTRY	POP	AREA	GDP	CONT	IND_DAY
2	CHN	China	1398.72	9596.96	12234.78	Asia	
3	IND	India	1351.16	3287.26	2575.67	Asia	1947-08-15
4	USA	US	329.74	9833.52	19485.39	N.America	1776-07-04
5	IDN	Indonesia	268.07	1910.93	1015.54	Asia	1945-08-17
6	BRA	Brazil	210.32	8515.77	2055.51	S.America	1822-09-07
7	PAK	Pakistan	205.71	881.91	302.14	Asia	1947-08-14
8	NGA	Nigeria	200.96	923.77	375.77	Africa	1960-10-01
9	BGD	Bangladesh	167.09	147.57	245.63	Asia	1971-03-26
10	RUS	Russia	146.79	17098.25	1530.75		1992-06-12
11	MEX	Mexico	126.58	1964.38	1158.23	N.America	1810-09-16
12	JPN	Japan	126.22	377.97	4872.42	Asia	
13	DEU	Germany	83.02	357.11	3693.2	Europe	
14	FRA	France	67.02	640.68	2582.49	Europe	1789-07-14
15	GBR	UK	66.44	242.5	2631.23	Europe	
16	ITA	Italy	60.36	301.34	1943.84	Europe	
17	ARG	Argentina	44.94	2780.4	637.49	S.America	1816-07-09
18	DZA	Algeria	43.38	2381.74	167.56	Africa	1962-07-05
19	CAN	Canada	37.59	9984.67	1647.12	N.America	1867-07-01
20	AUS	Australia	25.47	7692.02	1408.68	Oceania	
21	KAZ	Kazakhstan	18.53	2724.9	159.41	Asia	1991-12-16

`data.xlsx` viewed in Excel



	A	B	C	D	E	F	G
1		COUNTRY	POP	AREA	GDP	CONT	IND_DAY
2	CHN	China	1398.72	9596.96	12234.78	Asia	
3	IND	India	1351.16	3287.26	2575.67	Asia	1947-08-15
4	USA	US	329.74	9833.52	19485.39	N.America	1776-07-04
5	IDN	Indonesia	268.07	1910.93	1015.54	Asia	1945-08-17
6	BRA	Brazil	210.32	8515.77	2055.51	S.America	1822-09-07
7	PAK	Pakistan	205.71	881.91	302.14	Asia	1947-08-14
8	NGA	Nigeria	200.96	923.77	375.77	Africa	1960-10-01
9	BGD	Bangladesh	167.09	147.57	245.63	Asia	1971-03-26
10	RUS	Russia	146.79	17098.25	1530.75		1992-06-12
11	MEX	Mexico	126.58	1964.38	1158.23	N.America	1810-09-16
12	JPN	Japan	126.22	377.97	4872.42	Asia	
13	DEU	Germany	83.02	357.11	3693.2	Europe	
14	FRA	France	67.02	640.68	2582.49	Europe	1789-07-14
15	GBR	UK	66.44	242.5	2631.23	Europe	
16	ITA	Italy	60.36	301.34	1943.84	Europe	
17	ARG	Argentina	44.94	2780.4	637.49	S.America	1816-07-09
18	DZA	Algeria	43.38	2381.74	167.56	Africa	1962-07-05
19	CAN	Canada	37.59	9984.67	1647.12	N.America	1867-07-01
20	AUS	Australia	25.47	7692.02	1408.68	Oceania	
21	KAZ	Kazakhstan	18.53	2724.9	159.41	Asia	1991-12-16

`data.xlsx` viewed in Excel



	A	B	C	D	E	F	G
1		COUNTRY	POP	AREA	GDP	CONT	IND_DAY
2	CHN	China	1398.72	9596.96	12234.78	Asia	
3	IND	India	1351.16	3287.26	2575.67	Asia	1947-08-15
4	USA	US	329.74	9833.52	19485.39	N.America	1776-07-04
5	IDN	Indonesia	268.07	1910.93	1015.54	Asia	1945-08-17
6	BRA	Brazil	210.32	8515.77	2055.51	S.America	1822-09-07
7	PAK	Pakistan	205.71	881.91	302.14	Asia	1947-08-14
8	NGA	Nigeria	200.96	923.77	375.77	Africa	1960-10-01
9	BGD	Bangladesh	167.09	147.57	245.63	Asia	1971-03-26
10	RUS	Russia	146.79	17098.25	1530.75		1992-06-12
11	MEX	Mexico	126.58	1964.38	1158.23	N.America	1810-09-16
12	JPN	Japan	126.22	377.97	4872.42	Asia	
13	DEU	Germany	83.02	357.11	3693.2	Europe	
14	FRA	France	67.02	640.68	2582.49	Europe	1789-07-14
15	GBR	UK	66.44	242.5	2631.23	Europe	
16	ITA	Italy	60.36	301.34	1943.84	Europe	
17	ARG	Argentina	44.94	2780.4	637.49	S.America	1816-07-09
18	DZA	Algeria	43.38	2381.74	167.56	Africa	1962-07-05
19	CAN	Canada	37.59	9984.67	1647.12	N.America	1867-07-01
20	AUS	Australia	25.47	7692.02	1408.68	Oceania	
21	KAZ	Kazakhstan	18.53	2724.9	159.41	Asia	1991-12-16

Using Pandas to Read Large Excel Files in Python

https://realpython.com/working-with-large-excel-files-in-pandas/

Next: Working With Different File Types

Pandas: How to Read and Write Files

1. How To Read and Write Files
2. **Working With Different File Types**
   - 2.1 Understanding the Pandas IO API
   - 2.2 Working with CSV Files
   - 2.3 Working with JSON Files
   - 2.4 Working with HTML Files
   - 2.5 Working with Excel Files
   - 2.6 Working with SQL Files
   - 2.7 Working with Pickle Files
3. Working With Big Data

Working with Different File Types

- `.csv`
- Excel
- `.json`
- `.html`
- SQLite
- `.pickle`

Next: The Pandas IO API

Pandas: How to Read and Write Files

Pandas IO Tools

- Saves contents of `Series` and `DataFrame` objects
- Loads data from clipboard, objects or files

Understanding the Pandas IO API: Writing Files

Understanding the Pandas IO API: Writing Files

Use the pattern `.to_<file-type>()`

Understanding the Pandas IO API: Writing Files

- `.to_csv()`
- `.to_excel()`
- `.to_json()`
- `.to_html()`
- `.to_sql()`
- `.to_pickle()`

Understanding the Pandas IO API: Writing Files

Use the pattern

```
df.to_<file-type>(path_or_buf)
```

Understanding the Pandas IO API: Writing Files

Omitting the Path or Buffer:

```
>>> df.to_csv()
',IND\nCOUNTRY,India\nPOP,1351.16\nAREA,3287.26\n'
```

Understanding the Pandas IO API: Reading Files

Understanding the Pandas IO API: Reading Files

Use the pattern

```
pd.read_<file-type>()
```

Understanding the Pandas IO API: Reading Files

- `.read_csv()`
- `.read_excel()`
- `.read_json()`
- `.read_html()`
- `.read_sql()`
- `.read_pickle()`

Understanding the Pandas IO API: Reading Files

Valid Path or Buffer Examples

```
'data.csv'
'/home/users/realpython/data.xlsx'
'http://example.com/files/data.csv'
WindowsPath('c:/data/data.json')
```

Next: Working with CSV Files

Pandas: How to Read and Write Files

1. How To Read and Write Files
2. Working With Different File Types
   - 2.1 Understanding the Pandas IO API
   - 2.2 **Working with CSV Files**
   - 2.3 Working with JSON Files
   - 2.4 Working with HTML Files
   - 2.5 Working with Excel Files
   - 2.6 Working with SQL Files
   - 2.7 Working with Pickle Files
3. Working With Big Data

CSV Files

`nan` is Not A Number

- Non-signalling IEEE754 “Not A Number” Value

Generating `nan`

- `float('nan')`
- `math.nan`
- `numpy.nan`

new-data.csv

```
,COUNTRY,POP,AREA,GDP,CONT,IND_DAY
CHN,China,1398.72,9596.96,12234.78,Asia,(missing)
IND,India,1351.16,3287.26,2575.67,Asia,1947-08-15
USA,US,329.74,9833.52,19485.39,N.America,1776-07-04
IDN,Indonesia,268.07,1910.93,1015.54,Asia,1945-08-17
BRA,Brazil,210.32,8515.77,2055.51,S.America,1822-09-07
PAK,Pakistan,205.71,881.91,302.14,Asia,1947-08-14
NGA,Nigeria,200.96,923.77,375.77,Africa,1960-10-01
BGD,Bangladesh,167.09,147.57,245.63,Asia,1971-03-26
RUS,Russia,146.79,17098.25,1530.75,(missing),1992-06-12
MEX,Mexico,126.58,1964.38,1158.23,N.America,1810-09-16
JPN,Japan,126.22,377.97,4872.42,Asia,(missing)
DEU,Germany,83.02,357.11,3693.2,Europe,(missing)
FRA,France,67.02,640.68,2582.49,Europe,1789-07-14
GBR,UK,66.44,242.5,2631.23,Europe,(missing)
ITA,Italy,60.36,301.34,1943.84,Europe,(missing)
ARG,Argentina,44.94,2780.4,637.49,S.America,1816-07-09
DZA,Algeria,43.38,2381.74,167.56,Africa,1962-07-05
CAN,Canada,37.59,9984.67,1647.12,N.America,1867-07-01
AUS,Australia,25.47,7692.02,1408.68,Oceania,(missing)
KAZ,Kazakhstan,18.53,2724.9,159.41,Asia,1991-12-16
```

Missing Values

- `nan`
- `-nan`
- `NA`
- `N/A`
- `NaN`
- `null`

Optional Parameters

- `sep` - Value Separator
- `decimal` - Decimal Separator
- `encoding` - File Encoding
- `header` - Column Labels (`True` / `False`)

`.read_csv()` has Many Options!

- Managing Missing Data
- Working with Dates and Times
- Quoting
- Encoding
- Handling Errors
- … and more!

Using `.read_csv()` to obtain a Pandas `Series` Object

```
>>> ser = pd.read_csv('single.csv', squeeze=True)
>>> ser
0      single
1      column
2          of
3     entries
4          in
5        text
Name: a, dtype: object
```

Next: Working with JSON Files

Pandas: How to Read and Write Files

1. How To Read and Write Files
2. Working With Different File Types
   - 2.1 Understanding the Pandas IO API
   - 2.2 Working with CSV Files
   - 2.3 **Working with JSON Files**
   - 2.4 Working with HTML Files
   - 2.5 Working with Excel Files
   - 2.6 Working with SQL Files
   - 2.7 Working with Pickle Files
3. Working With Big Data

`.json` Files

- **J**ava**S**cript **O**bject **N**otation
- Plaintext
- Human-Readable
- ISO/IEC 21778:2017 and ECMA-404
- `.json` Extension
- Built-in Python Support with `json` Library

data-columns.json


```
{
 "COUNTRY":{"CHN":"China","IND":"India","USA":"US","IDN":"Indonesia","BRA":"Brazil", ...},
 "POP":{"CHN":1398.72,"IND":1351.16,"USA":329.74,"IDN":268.07,"BRA":210.32, ...},
 "AREA":{"CHN":9596.96,"IND":3287.26,"USA":9833.52,"IDN":1910.93,"BRA":8515.77, ...},
 "GDP":{"CHN":12234.78,"IND":2575.67,"USA":19485.39,"IDN":1015.54,"BRA":2055.51, ...},
 "CONT":{"CHN":"Asia","IND":"Asia","USA":"N.America","IDN":"Asia","BRA":"S.America", ...},
 "IND_DAY":{"CHN":null,"IND":"1947-08-15","USA":"1776-07-04","IDN":"1945-08-17", ...},
}
```


Note: Lines have been shortened for clarity

data-index.json


```
{
 "CHN":{"COUNTRY":"China","POP":1398.72,"AREA":9596.96,"GDP":12234.78,"CONT":"Asia","IND_DAY":null},
 "IND":{"COUNTRY":"India","POP":1351.16,"AREA":3287.26,"GDP":2575.67,"CONT":"Asia","IND_DAY":"1947-08-15"},
 "USA":{"COUNTRY":"US","POP":329.74,"AREA":9833.52,"GDP":19485.39,"CONT":"N.America","IND_DAY":"1776-07-04"},
 "IDN":{"COUNTRY":"Indonesia","POP":268.07,"AREA":1910.93,"GDP":1015.54,"CONT":"Asia","IND_DAY":"1945-08-17"},
 "BRA":{"COUNTRY":"Brazil","POP":210.32,"AREA":8515.77,"GDP":2055.51,"CONT":"S.America","IND_DAY":"1822-09-07"},
 "PAK":{"COUNTRY":"Pakistan","POP":205.71,"AREA":881.91,"GDP":302.14,"CONT":"Asia","IND_DAY":"1947-08-14"},
 "NGA":{"COUNTRY":"Nigeria","POP":200.96,"AREA":923.77,"GDP":375.77,"CONT":"Africa","IND_DAY":"1960-10-01"},
 "BGD":{"COUNTRY":"Bangladesh","POP":167.09,"AREA":147.57,"GDP":245.63,"CONT":"Asia","IND_DAY":"1971-03-26"},
 "RUS":{"COUNTRY":"Russia","POP":146.79,"AREA":17098.25,"GDP":1530.75,"CONT":null,"IND_DAY":"1992-06-12"},
...
}
```


Note: Lines have been removed for clarity

data-records.json


```
[
 {"COUNTRY":"China","POP":1398.72,"AREA":9596.96,"GDP":12234.78,"CONT":"Asia","IND_DAY":null},
 {"COUNTRY":"India","POP":1351.16,"AREA":3287.26,"GDP":2575.67,"CONT":"Asia","IND_DAY":"1947-08-15"},
 {"COUNTRY":"US","POP":329.74,"AREA":9833.52,"GDP":19485.39,"CONT":"N.America","IND_DAY":"1776-07-04"},
 {"COUNTRY":"Indonesia","POP":268.07,"AREA":1910.93,"GDP":1015.54,"CONT":"Asia","IND_DAY":"1945-08-17"},
 {"COUNTRY":"Brazil","POP":210.32,"AREA":8515.77,"GDP":2055.51,"CONT":"S.America","IND_DAY":"1822-09-07"},
 {"COUNTRY":"Pakistan","POP":205.71,"AREA":881.91,"GDP":302.14,"CONT":"Asia","IND_DAY":"1947-08-14"},
 {"COUNTRY":"Nigeria","POP":200.96,"AREA":923.77,"GDP":375.77,"CONT":"Africa","IND_DAY":"1960-10-01"},
 {"COUNTRY":"Bangladesh","POP":167.09,"AREA":147.57,"GDP":245.63,"CONT":"Asia","IND_DAY":"1971-03-26"},
 {"COUNTRY":"Russia","POP":146.79,"AREA":17098.25,"GDP":1530.75,"CONT":null,"IND_DAY":"1992-06-12"},
...
]
```


Note: Lines have been removed for clarity

data-split.json

```
{
 "columns":["COUNTRY","POP","AREA","GDP","CONT","IND_DAY"],
 "index":["CHN","IND","USA","IDN","BRA","PAK","NGA","BGD","RUS",  "MEX","JPN","DEU","FRA","GBR","ITA","ARG","DZA", ...],
 "data":[
  ["China",1398.72,9596.96,12234.78,"Asia",null],
  ["India",1351.16,3287.26,2575.67,"Asia","1947-08-15"],
  ["US",329.74,9833.52,19485.39,"N.America","1776-07-04"],
  ["Indonesia",268.07,1910.93,1015.54,"Asia","1945-08-17"],
  ["Brazil",210.32,8515.77,2055.51,"S.America","1822-09-07"],
  ["Pakistan",205.71,881.91,302.14,"Asia","1947-08-14"],
  ["Nigeria",200.96,923.77,375.77,"Africa","1960-10-01"],
  ...
  ]
}
```

Note: Lines have been removed and shortened for clarity

Omitting `path_or_buf`

```
>>> df.to_json()
'{"COUNTRY":{"CHN":"China","IND":"India","USA":"US","IDN":"Indonesia","BRA":"Brazil","PAK":"Pakistan","NGA":"Nigeria","BGD":
"Bangladesh","RUS":"Russia","MEX":"Mexico","JPN":"Japan","DEU":"Germany","FRA":"France","GBR":"UK","ITA":"Italy","ARG":"Arge
ntina","DZA":"Algeria","CAN":"Canada","AUS":"Australia","KAZ":"Kazakhstan"},"POP":{"CHN":1398.72,"IND":1351.16,"USA":329.74,
"IDN":268.07,"BRA":210.32,"PAK":205.71,"NGA":200.96,"BGD":167.09,"RUS":146.79,"MEX":126.58,"JPN":126.22,"DEU":83.02,"FRA":67
.02,"GBR":66.44,"ITA":60.36,"ARG":44.94,"DZA":43.38,"CAN":37.59,"AUS":25.47,"KAZ":18.53},"AREA":{"CHN":9596.96,"IND":3287.26
,"USA":9833.52,"IDN":1910.93,"BRA":8515.77,"PAK":881.91,"NGA":923.77,"BGD":147.57,"RUS":17098.25,"MEX":1964.38,"JPN":377.97,
"DEU":357.11,"FRA":640.68,"GBR":242.5,"ITA":301.34,"ARG":2780.4,"DZA":2381.74,"CAN":9984.67,"AUS":7692.02,"KAZ":2724.9},"GDP
":{"CHN":12234.78,"IND":2575.67,"USA":19485.39,"IDN":1015.54,"BRA":2055.51,"PAK":302.14,"NGA":375.77,"BGD":245.63,"RUS":1530
.75,"MEX":1158.23,"JPN":4872.42,"DEU":3693.2,"FRA":2582.49,"GBR":2631.23,"ITA":1943.84,"ARG":637.49,"DZA":167.56,"CAN":1647.
12,"AUS":1408.68,"KAZ":159.41},"CONT":{"CHN":"Asia","IND":"Asia","USA":"N.America","IDN":"Asia","BRA":"S.America","PAK":"Asi
a","NGA":"Africa","BGD":"Asia","RUS":null,"MEX":"N.America","JPN":"Asia","DEU":"Europe","FRA":"Europe","GBR":"Europe","ITA":
"Europe","ARG":"S.America","DZA":"Africa","CAN":"N.America","AUS":"Oceania","KAZ":"Asia"},"IND_DAY":{"CHN":null,"IND":-70632
0000000,"USA":-6106060800000,"IDN":-769219200000,"BRA":-4648924800000,"PAK":-706406400000,"NGA":-291945600000,"BGD":38793600
000,"RUS":708307200000,"MEX":-5026838400000,"JPN":null,"DEU":null,"FRA":-5694969600000,"GBR":null,"ITA":null,"ARG":-48434112
00000,"DZA":-236476800000,"CAN":-3234729600000,"AUS":null,"KAZ":692841600000}}'
```

`.to_json()` Optional Parameters

- `index=False` - Row Labels are not Saved
- `double_precision` - The Number of Decimal Places to Use
- `date_format` - Controls Date Format (`epoch` or `iso`)
- `date_unit` - Controls Date Resolution (**s**econds to **n**ano**s**econds)

data-time.json


```
{
 "COUNTRY":{"CHN":"China","IND":"India","USA":"US","IDN":"Indonesia","BRA":"Brazil", ...},
 "POP":{"CHN":1398.72,"IND":1351.16,"USA":329.74,"IDN":268.07,"BRA":210.32, ...},
 "AREA":{"CHN":9596.96,"IND":3287.26,"USA":9833.52,"IDN":1910.93,"BRA":8515.77, ...},
 "GDP":{"CHN":12234.78,"IND":2575.67,"USA":19485.39,"IDN":1015.54,"BRA":2055.51, ...},
 "CONT":{"CHN":"Asia","IND":"Asia","USA":"N.America","IDN":"Asia","BRA":"S.America", ...},
 "IND_DAY":{"CHN":null,"IND":-706320000000,"USA":-610606080000,"IDN":-769219200000, ...}
}
```


Note: Lines have been shortened for clarity

new-data-time.json


```
{
 "COUNTRY":{"CHN":"China","IND":"India","USA":"US","IDN":"Indonesia","BRA":"Brazil", ...},
 "POP":{"CHN":1398.72,"IND":1351.16,"USA":329.74,"IDN":268.07,"BRA":210.32, ...},
 "AREA":{"CHN":9596.96,"IND":3287.26,"USA":9833.52,"IDN":1910.93,"BRA":8515.77, ...},
 "GDP":{"CHN":12234.78,"IND":2575.67,"USA":19485.39,"IDN":1015.54,"BRA":2055.51, ...},
 "CONT":{"CHN":"Asia","IND":"Asia","USA":"N.America","IDN":"Asia","BRA":"S.America", ...},
 "IND_DAY":{"CHN":null,"IND":"1947-08-15T00:00:00Z","USA":"1776-07-04T00:00:00Z", ...}
}
```


Note: Lines have been shortened for clarity

`.read_json()` Optional Parameters

- `encoding` - Set Encoding
- `convert_dates` and `keep_default_dates` to Manipulate Dates
- `dtype` and `precise_float` to Control Precision
- `numpy=True` to decode directly to NumPy Arrays

JSON May Not Preserve Row and Column Order

Next: HTML Files

Pandas: How to Read and Write Files

1. How To Read and Write Files
2. Working With Different File Types
   - 2.1 Understanding the Pandas IO API
   - 2.2 Working with CSV Files
   - 2.3 Working with JSON Files
   - 2.4 **Working with HTML Files**
   - 2.5 Working with Excel Files
   - 2.6 Working with SQL Files
   - 2.7 Working with Pickle Files
3. Working With Big Data

HTML

- Plaintext
- Uses Hypertext Markup Language
- `.html` and `.htm` Filename Extensions
- An HTML Parser Library is Needed

`data.html` viewed in a browser

	COUNTRY	POP	AREA	GDP	CONT	IND_DAY
CHN	China	1398.72	9596.96	12234.78	Asia	NaT
IND	India	1351.16	3287.26	2575.67	Asia	1947-08-15
USA	US	329.74	9833.52	19485.39	N.America	1776-07-04
IDN	Indonesia	268.07	1910.93	1015.54	Asia	1945-08-17
BRA	Brazil	210.32	8515.77	2055.51	S.America	1822-09-07
PAK	Pakistan	205.71	881.91	302.14	Asia	1947-08-14
NGA	Nigeria	200.96	923.77	375.77	Africa	1960-10-01
BGD	Bangladesh	167.09	147.57	245.63	Asia	1971-03-26
RUS	Russia	146.79	17098.25	1530.75	NaN	1992-06-12
MEX	Mexico	126.58	1964.38	1158.23	N.America	1810-09-16
JPN	Japan	126.22	377.97	4872.42	Asia	NaT
DEU	Germany	83.02	357.11	3693.2	Europe	NaT
FRA	France	67.02	640.68	2582.49	Europe	1789-07-14
GBR	UK	66.44	242.5	2631.23	Europe	NaT
ITA	Italy	60.36	301.34	1943.84	Europe	NaT
ARG	Argentina	44.94	2780.4	637.49	S.America	1816-07-09
DZA	Algeria	43.38	2381.74	167.56	Africa	1962-07-05
CAN	Canada	37.59	9984.67	1647.12	N.America	1867-07-01
AUS	Australia	25.47	7692.02	1408.68	Oceania	NaT
KAZ	Kazakhstan	18.53	2724.9	159.41	Asia	1991-12-16

`data.html` code

```
<table border="1" class="dataframe">
 <thead>
 <tr style="text-align: right;">
 <th></th><th>COUNTRY</th><th>POP</th><th>AREA</th><th>GDP</th><th>CONT</th><th>IND_DAY</th>
 </tr>
 </thead>
 <tbody>
 <tr>
 <th>CHN</th><td>China</td><td>1398.72</td><td>9596.96</td><td>12234.78</td><td>Asia</td><td>NaT</td>
 </tr>
 </tbody>
</table>
```

Note: Lines have been removed for clarity

`.to_html()` without `path_or_buf`

```
>>> df.to_html()
'<table border="1" class="dataframe">\n  <thead>\n    <tr style="text-align: right;">\n
      <th></th>\n      <th>COUNTRY</th>\n      <th>POP</th>\n      <th>AREA</th>\n
      <th>GDP</th>\n      <th>CONT</th>\n      <th>IND_DAY</th>\n    </tr>\n
      </thead>\n  <tbody>\n    <tr>\n    <th>CHN</th>\n      <td>China</td>\n
      <td>1398.72</td>\n      <td>9596.96</td>\n      <td>12234.78</td>\n
      <td>Asia</td>\n      <td>NaN</td>\n    </tr>\n    <tr>\n      <th>IND</th>\n
      <td>India</td>\n      <td>1351.16</td>\n      <td>3287.26</td>\n      <td>2575.67</td>\n
      <td>Asia</td>\n      <td>1947-08-15</td>\n    </tr>\n    <tr>\n      <th>USA</th>\n
      <td>US</td>\n      <td>329.74</td>\n      <td>9833.52</td>\n      <td>19485.39</td>\n
      ...
```

Note: Lines have been removed for clarity

`.to_html()` Optional Parameters

- `header` - Controls Saving of Column Names
- `index` - Controls Saving of Row Labels
- `classes` - Assigns CSS Classes
- `render_links` - Controls Conversion of URLs to HTML Links
- `table_id` - Assigns CSS ID of Table
- `escape` - Controls Conversion of `<`, `>` and `&` to HTML-safe Strings

`.read_html()` Optional Parameters

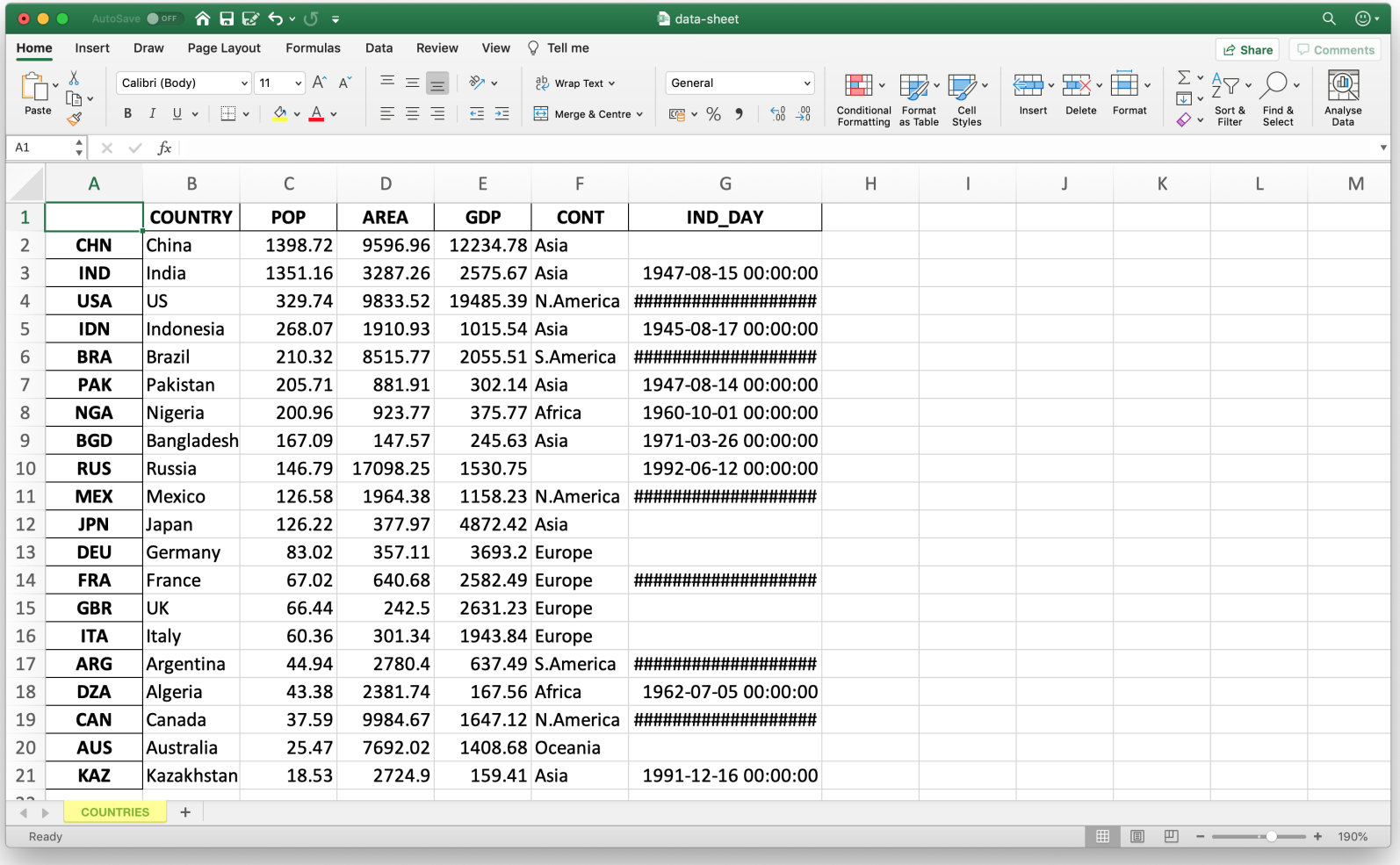
- `parse_dates` - Controls Interpretation of Columns Containing Dates
- `na_values` - Custom NA Values
- `encoding` - Defines Encoding used to Decode Web Page
- `flavor` - Parsing Engine to Use

Next: Excel Files

Pandas: How to Read and Write Files

Excel Files

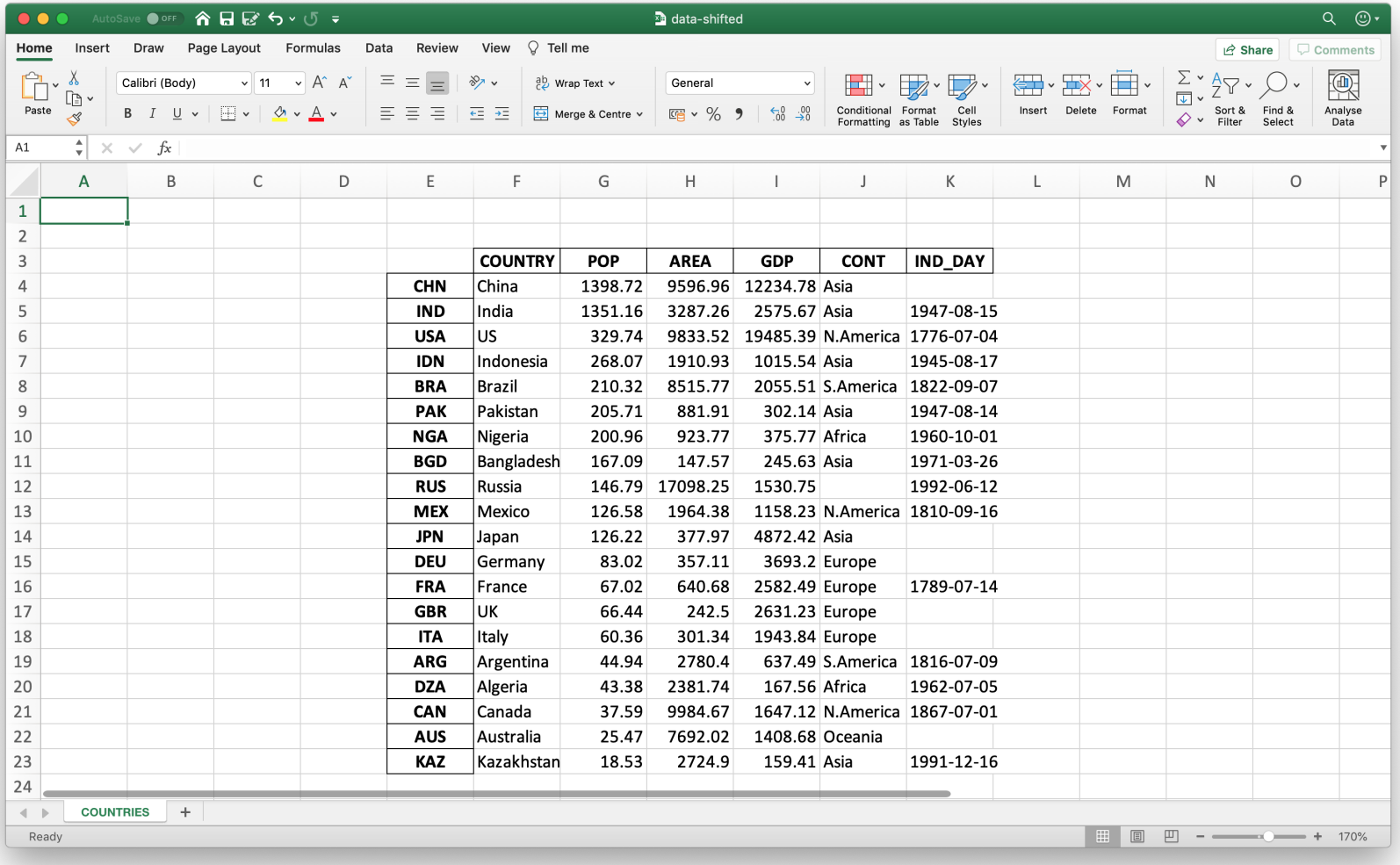
`data-sheet.xlsx` viewed in Excel



data-sheet

	A	B	C	D	E	F	G
1		COUNTRY	POP	AREA	GDP	CONT	IND_DAY
2	CHN	China	1398.72	9596.96	12234.78	Asia	
3	IND	India	1351.16	3287.26	2575.67	Asia	1947-08-15 00:00:00
4	USA	US	329.74	9833.52	19485.39	N.America	##########
5	IDN	Indonesia	268.07	1910.93	1015.54	Asia	1945-08-17 00:00:00
6	BRA	Brazil	210.32	8515.77	2055.51	S.America	##########
7	PAK	Pakistan	205.71	881.91	302.14	Asia	1947-08-14 00:00:00
8	NGA	Nigeria	200.96	923.77	375.77	Africa	1960-10-01 00:00:00
9	BGD	Bangladesh	167.09	147.57	245.63	Asia	1971-03-26 00:00:00
10	RUS	Russia	146.79	17098.25	1530.75		1992-06-12 00:00:00
11	MEX	Mexico	126.58	1964.38	1158.23	N.America	##########
12	JPN	Japan	126.22	377.97	4872.42	Asia	
13	DEU	Germany	83.02	357.11	3693.2	Europe	
14	FRA	France	67.02	640.68	2582.49	Europe	##########
15	GBR	UK	66.44	242.5	2631.23	Europe	
16	ITA	Italy	60.36	301.34	1943.84	Europe	
17	ARG	Argentina	44.94	2780.4	637.49	S.America	##########
18	DZA	Algeria	43.38	2381.74	167.56	Africa	1962-07-05 00:00:00
19	CAN	Canada	37.59	9984.67	1647.12	N.America	##########
20	AUS	Australia	25.47	7692.02	1408.68	Oceania	
21	KAZ	Kazakhstan	18.53	2724.9	159.41	Asia	1991-12-16 00:00:00

`data-shifted.xlsx` viewed in Excel



	COUNTRY	POP	AREA	GDP	CONT	IND_DAY
CHN	China	1398.72	9596.96	12234.78	Asia	
IND	India	1351.16	3287.26	2575.67	Asia	1947-08-15
USA	US	329.74	9833.52	19485.39	N.America	1776-07-04
IDN	Indonesia	268.07	1910.93	1015.54	Asia	1945-08-17
BRA	Brazil	210.32	8515.77	2055.51	S.America	1822-09-07
PAK	Pakistan	205.71	881.91	302.14	Asia	1947-08-14
NGA	Nigeria	200.96	923.77	375.77	Africa	1960-10-01
BGD	Bangladesh	167.09	147.57	245.63	Asia	1971-03-26
RUS	Russia	146.79	17098.25	1530.75		1992-06-12
MEX	Mexico	126.58	1964.38	1158.23	N.America	1810-09-16
JPN	Japan	126.22	377.97	4872.42	Asia	
DEU	Germany	83.02	357.11	3693.2	Europe	
FRA	France	67.02	640.68	2582.49	Europe	1789-07-14
GBR	UK	66.44	242.5	2631.23	Europe	
ITA	Italy	60.36	301.34	1943.84	Europe	
ARG	Argentina	44.94	2780.4	637.49	S.America	1816-07-09
DZA	Algeria	43.38	2381.74	167.56	Africa	1962-07-05
CAN	Canada	37.59	9984.67	1647.12	N.America	1867-07-01
AUS	Australia	25.47	7692.02	1408.68	Oceania	
KAZ	Kazakhstan	18.53	2724.9	159.41	Asia	1991-12-16

`.read_excel()` - `sheet_name`

- Zero-Based Index of Worksheet - `sheet_name=0`
- Worksheet Name - `sheet_name="COUNTRIES"`
- List of Indices or Names - `sheet_name=[0,1,3]`
- `None` to read all sheets - `sheet_name=None`

`.read_excel()` Optional Parameters

- `engine` - `“xlrd”`, `“openpyxl”`, `“odf”`, `“pyxlsb”`
- `dtype` - Specify Data Types
- `na_values` - Specify strings to recognize as `nan`
- `parse_dates` - Control Parsing of Columns as Dates

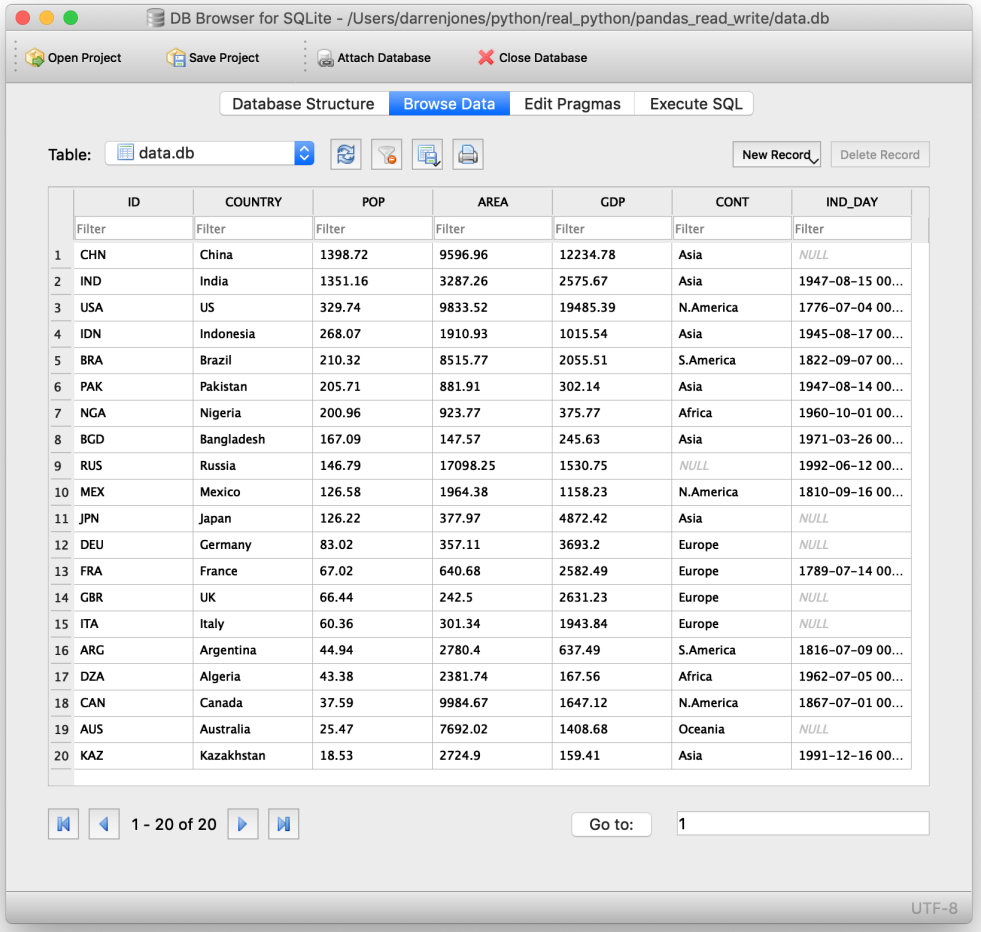
Next: SQL Files

Pandas: How to Read and Write Files

1. How To Read and Write Files
2. Working With Different File Types
   - 2.1 Understanding the Pandas IO API
   - 2.2 Working with CSV Files
   - 2.3 Working with JSON Files
   - 2.4 Working with HTML Files
   - 2.5 Working with Excel Files
   - 2.6 **Working with SQL Files**
   - 2.7 Working with Pickle Files
3. Working With Big Data

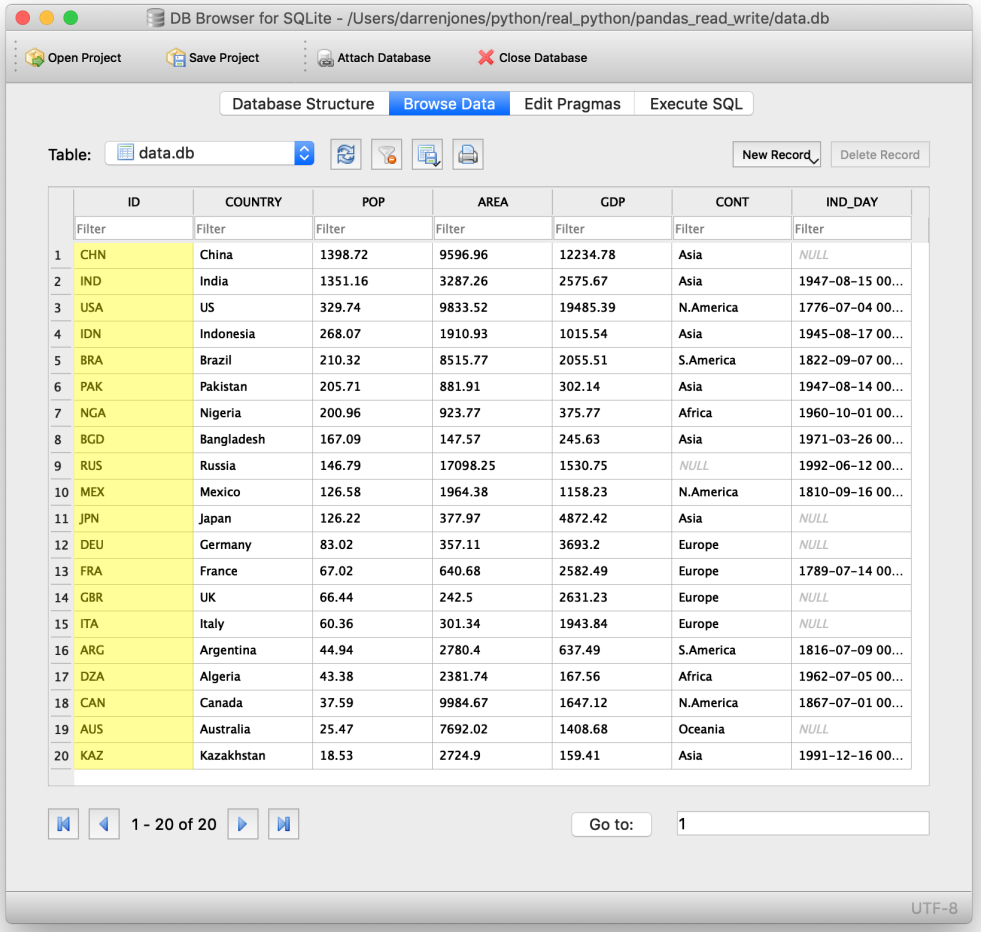
SQL

`data.db` viewed in a SQLite Viewer



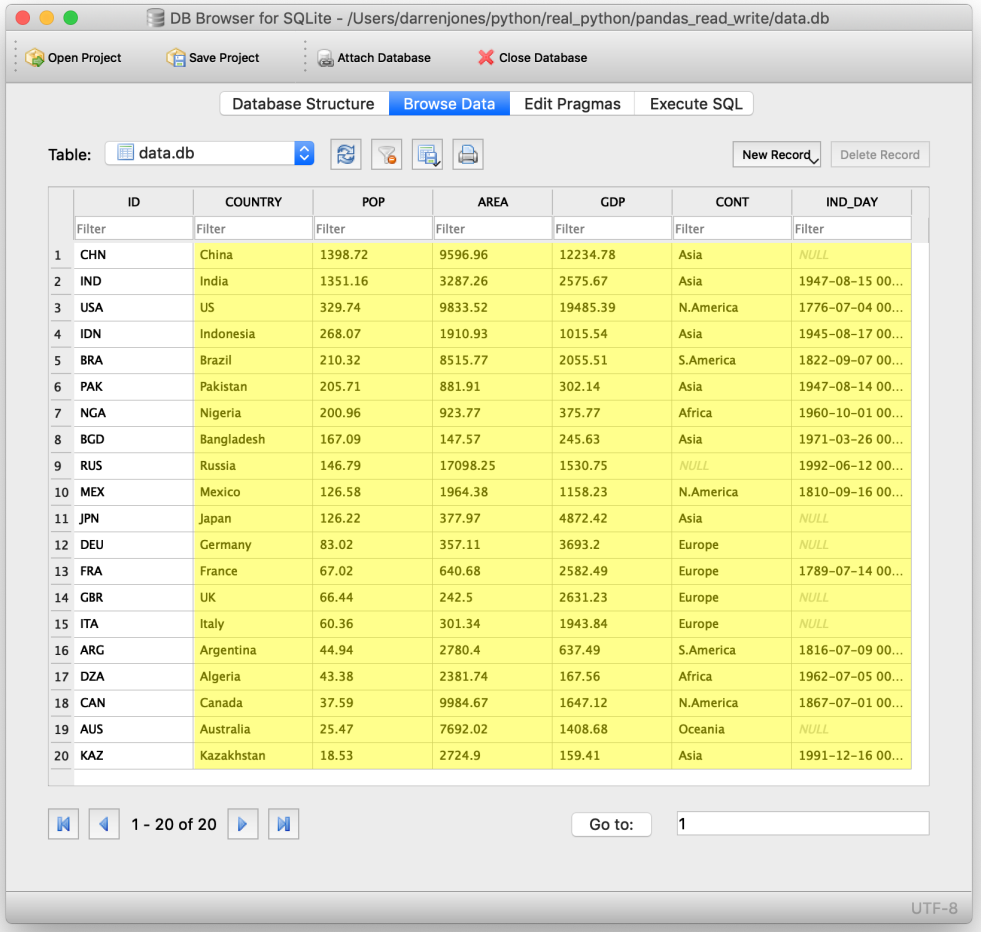
	ID	COUNTRY	POP	AREA	GDP	CONT	IND_DAY
	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	CHN	China	1398.72	9596.96	12234.78	Asia	NULL
2	IND	India	1351.16	3287.26	2575.67	Asia	1947-08-15 00...
3	USA	US	329.74	9833.52	19485.39	N.America	1776-07-04 00...
4	IDN	Indonesia	268.07	1910.93	1015.54	Asia	1945-08-17 00...
5	BRA	Brazil	210.32	8515.77	2055.51	S.America	1822-09-07 00...
6	PAK	Pakistan	205.71	881.91	302.14	Asia	1947-08-14 00...
7	NGA	Nigeria	200.96	923.77	375.77	Africa	1960-10-01 00...
8	BGD	Bangladesh	167.09	147.57	245.63	Asia	1971-03-26 00...
9	RUS	Russia	146.79	17098.25	1530.75	NULL	1992-06-12 00...
10	MEX	Mexico	126.58	1964.38	1158.23	N.America	1810-09-16 00...
11	JPN	Japan	126.22	377.97	4872.42	Asia	NULL
12	DEU	Germany	83.02	357.11	3693.2	Europe	NULL
13	FRA	France	67.02	640.68	2582.49	Europe	1789-07-14 00...
14	GBR	UK	66.44	242.5	2631.23	Europe	NULL
15	ITA	Italy	60.36	301.34	1943.84	Europe	NULL
16	ARG	Argentina	44.94	2780.4	637.49	S.America	1816-07-09 00...
17	DZA	Algeria	43.38	2381.74	167.56	Africa	1962-07-05 00...
18	CAN	Canada	37.59	9984.67	1647.12	N.America	1867-07-01 00...
19	AUS	Australia	25.47	7692.02	1408.68	Oceania	NULL
20	KAZ	Kazakhstan	18.53	2724.9	159.41	Asia	1991-12-16 00...

`data.db` first column contains row labels



	ID	COUNTRY	POP	AREA	GDP	CONT	IND_DAY
1	CHN	China	1398.72	9596.96	12234.78	Asia	NULL
2	IND	India	1351.16	3287.26	2575.67	Asia	1947-08-15 00...
3	USA	US	329.74	9833.52	19485.39	N.America	1776-07-04 00...
4	IDN	Indonesia	268.07	1910.93	1015.54	Asia	1945-08-17 00...
5	BRA	Brazil	210.32	8515.77	2055.51	S.America	1822-09-07 00...
6	PAK	Pakistan	205.71	881.91	302.14	Asia	1947-08-14 00...
7	NGA	Nigeria	200.96	923.77	375.77	Africa	1960-10-01 00...
8	BGD	Bangladesh	167.09	147.57	245.63	Asia	1971-03-26 00...
9	RUS	Russia	146.79	17098.25	1530.75	NULL	1992-06-12 00...
10	MEX	Mexico	126.58	1964.38	1158.23	N.America	1810-09-16 00...
11	JPN	Japan	126.22	377.97	4872.42	Asia	NULL
12	DEU	Germany	83.02	357.11	3693.2	Europe	NULL
13	FRA	France	67.02	640.68	2582.49	Europe	1789-07-14 00...
14	GBR	UK	66.44	242.5	2631.23	Europe	NULL
15	ITA	Italy	60.36	301.34	1943.84	Europe	NULL
16	ARG	Argentina	44.94	2780.4	637.49	S.America	1816-07-09 00...
17	DZA	Algeria	43.38	2381.74	167.56	Africa	1962-07-05 00...
18	CAN	Canada	37.59	9984.67	1647.12	N.America	1867-07-01 00...
19	AUS	Australia	25.47	7692.02	1408.68	Oceania	NULL
20	KAZ	Kazakhstan	18.53	2724.9	159.41	Asia	1991-12-16 00...

`data.db` remaining columns contain data

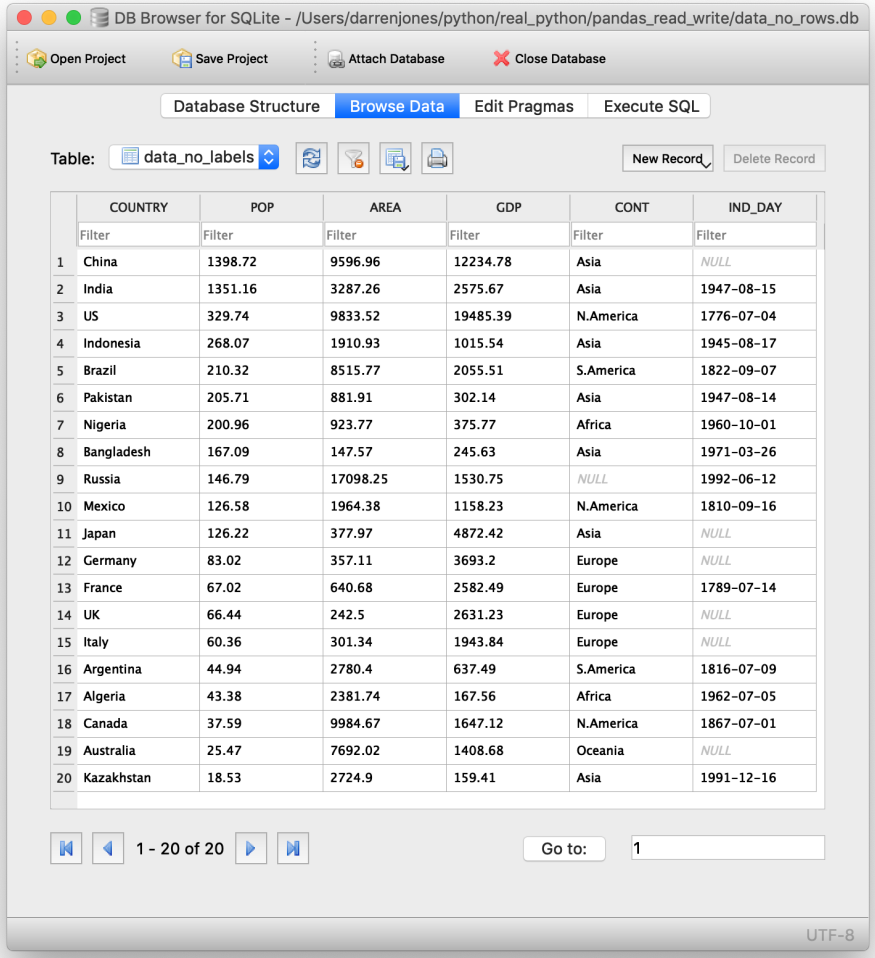


	ID	COUNTRY	POP	AREA	GDP	CONT	IND_DAY
	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	CHN	China	1398.72	9596.96	12234.78	Asia	NULL
2	IND	India	1351.16	3287.26	2575.67	Asia	1947-08-15 00...
3	USA	US	329.74	9833.52	19485.39	N.America	1776-07-04 00...
4	IDN	Indonesia	268.07	1910.93	1015.54	Asia	1945-08-17 00...
5	BRA	Brazil	210.32	8515.77	2055.51	S.America	1822-09-07 00...
6	PAK	Pakistan	205.71	881.91	302.14	Asia	1947-08-14 00...
7	NGA	Nigeria	200.96	923.77	375.77	Africa	1960-10-01 00...
8	BGD	Bangladesh	167.09	147.57	245.63	Asia	1971-03-26 00...
9	RUS	Russia	146.79	17098.25	1530.75	NULL	1992-06-12 00...
10	MEX	Mexico	126.58	1964.38	1158.23	N.America	1810-09-16 00...
11	JPN	Japan	126.22	377.97	4872.42	Asia	NULL
12	DEU	Germany	83.02	357.11	3693.2	Europe	NULL
13	FRA	France	67.02	640.68	2582.49	Europe	1789-07-14 00...
14	GBR	UK	66.44	242.5	2631.23	Europe	NULL
15	ITA	Italy	60.36	301.34	1943.84	Europe	NULL
16	ARG	Argentina	44.94	2780.4	637.49	S.America	1816-07-09 00...
17	DZA	Algeria	43.38	2381.74	167.56	Africa	1962-07-05 00...
18	CAN	Canada	37.59	9984.67	1647.12	N.America	1867-07-01 00...
19	AUS	Australia	25.47	7692.02	1408.68	Oceania	NULL
20	KAZ	Kazakhstan	18.53	2724.9	159.41	Asia	1991-12-16 00...

Using `index=False` to omit row labels

```
>>> df.to_sql('data.db', con=engine, index=False)
```

`data_no_row_labels.db` viewed in a database browser



	COUNTRY	POP	AREA	GDP	CONT	IND_DAY
	Filter	Filter	Filter	Filter	Filter	Filter
1	China	1398.72	9596.96	12234.78	Asia	NULL
2	India	1351.16	3287.26	2575.67	Asia	1947-08-15
3	US	329.74	9833.52	19485.39	N.America	1776-07-04
4	Indonesia	268.07	1910.93	1015.54	Asia	1945-08-17
5	Brazil	210.32	8515.77	2055.51	S.America	1822-09-07
6	Pakistan	205.71	881.91	302.14	Asia	1947-08-14
7	Nigeria	200.96	923.77	375.77	Africa	1960-10-01
8	Bangladesh	167.09	147.57	245.63	Asia	1971-03-26
9	Russia	146.79	17098.25	1530.75	NULL	1992-06-12
10	Mexico	126.58	1964.38	1158.23	N.America	1810-09-16
11	Japan	126.22	377.97	4872.42	Asia	NULL
12	Germany	83.02	357.11	3693.2	Europe	NULL
13	France	67.02	640.68	2582.49	Europe	1789-07-14
14	UK	66.44	242.5	2631.23	Europe	NULL
15	Italy	60.36	301.34	1943.84	Europe	NULL
16	Argentina	44.94	2780.4	637.49	S.America	1816-07-09
17	Algeria	43.38	2381.74	167.56	Africa	1962-07-05
18	Canada	37.59	9984.67	1647.12	N.America	1867-07-01
19	Australia	25.47	7692.02	1408.68	Oceania	NULL
20	Kazakhstan	18.53	2724.9	159.41	Asia	1991-12-16

`.to_sql()` Optional Parameters

- `schema` - Specify Database Schema
- `dtype` - Specify Data Types
- `if_exists` - Specify Behaviour on Existing Database

`.to_sql()` - `if_exists` options

- `fail` - Raise a `ValueError` (default)
- `replace` - Drops the Existing Table and Inserts New Values
- `append` - Inserts New Values to the Existing Table

Next: Pickle Files

Pandas: How to Read and Write Files

1. How To Read and Write Files
2. Working With Different File Types
   - 2.1 Understanding the Pandas IO API
   - 2.2 Working with CSV Files
   - 2.3 Working with JSON Files
   - 2.4 Working with HTML Files
   - 2.5 Working with Excel Files
   - 2.6 Working with SQL Files
   - 2.7 **Working with Pickle Files**
3. Working With Big Data

Pickle Files

- Picking converts a Python Object to a Byte Stream
- Unpickling converts a Byte Stream to a Python Object
- Maintain the Data and Hierarchy of Python Objects
- `.pickle` or `.pkl` extension

Pickle Protocols

	Human-Readable	Python Version	Comments	Default
`0`	✅		All	Original Protocol
`1`	❌	All	First Binary Format	
`2`	❌	>= 2.3	More Efficent Storage	
`3`	❌		>= 3.0	Supports 3.0 Features
`4`	❌		>= 3.4	Multiple Improvements
`5`	❌	>= 3.8	Improved performance	

Take Care with Pickle Files!

- Loading from Untrusted Sources
- Unpicking Can Execute Arbitrary Code
- Can Give Access to Any Element of Your Machine

Next: Working with Big Data

Pandas: How to Read and Write Files

1. How To Read and Write Files
2. Working With Different File Types
3. **Working With Big Data**
   - 3.1 Compress
   - 3.2 Omit Columns
   - 3.3 Omit Rows
   - 3.4 Force Lower Precision
   - 3.5 Use Chunks

Working with Big Data

- Compress
- Choose
- Omit
- Force
- Split

Next: Using Compression

Pandas: How to Read and Write Files

1. How To Read and Write Files
2. Working With Different File Types
3. Working With Big Data
   - 3.1 **Use Compression**
   - 3.2 Choose Columns
   - 3.3 Omit Rows
   - 3.4 Force Lower Precision
   - 3.5 Use Chunks

Compress and Decompress Files

Compress and Decompress Files

- `.gz`
- `.bz2`
- `.zip`
- `.xz`

`compression` parameter values

- `infer`
- `gzip`
- `bz2`
- `zip`
- `xz`
- `none`

Next: Choosing Columns

Pandas: How to Read and Write Files

1. How To Read and Write Files
2. Working With Different File Types
3. Working With Big Data
   - 3.1 Use Compression
   - 3.2 **Choose Columns**
   - 3.3 Omit Rows
   - 3.4 Force Lower Precision
   - 3.5 Use Chunks

Choose Columns

- `.read_csv()` & `.read_excel()`
- `usecols` parameter
- List of Column Names or Indices

Next: Omitting Rows

Pandas: How to Read and Write Files

1. How To Read and Write Files
2. Working With Different File Types
3. Working With Big Data
   - 3.1 Use Compression
   - 3.2 Choose Columns
   - 3.3 **Omit Rows**
   - 3.4 Force Lower Precision
   - 3.5 Use Chunks

Omit Rows

- `.read_csv()` & `.read_excel()`
- `skiprows` - Number of Rows or List of Rows to Skip
- `skipfooter` - Number of Rows to Omit at the End
- `nrows` - Number of Rows to Read

Next: Forcing Less Precise Data Types

Pandas: How to Read and Write Files

1. How To Read and Write Files
2. Working With Different File Types
3. Working With Big Data
   - 3.1 Use Compression
   - 3.2 Choose Columns
   - 3.3 Omit Rows
   - 3.4 **Force Lower Precision**
   - 3.5 Use Chunks

Force Less Precise Data Types

Next: Using Chunks To Iterate Through Files

Pandas: How to Read and Write Files

1. How To Read and Write Files
2. Working With Different File Types
3. Working With Big Data
   - 3.1 Use Compression
   - 3.2 Choose Columns
   - 3.3 Omit Rows
   - 3.4 Force Lower Precision
   - 3.5 **Use Chunks**

Use Chunks To Iterate Through Files

- `.read_csv()`
- `.read_json()`
- `.read_sql()`
- Optional Parameter `chunksize`

Use Chunks To Iterate Through Files

- Each Iteration Processes `chunksize` Rows
- Last Iteration may be Smaller Than `chunksize`
- Manage Required Memory

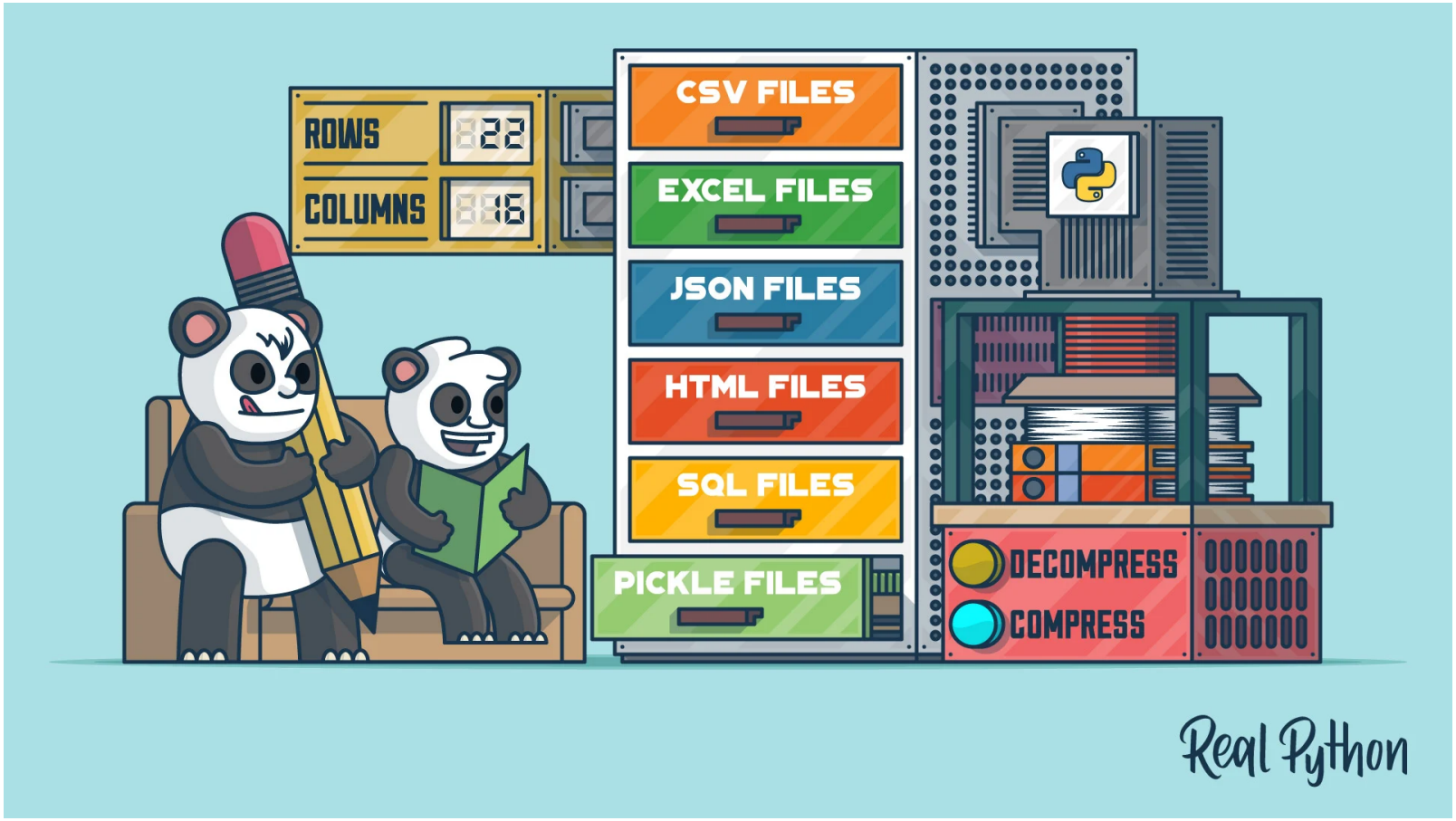
Next: Summary

PANDAS: HOW TO READ AND WRITE FILES: SUMMARY

Summary

- Save data from `DataFrame` objects
- Load data into `DataFrame` objects
- `.read_csv()` and `.to_csv()`
- Excel, JSON, HTML, SQL and Pickle
- Techniques for Big Data

Summary


ROWS
22
COLUMNS
16
CSV FILES
EXCEL FILES
JSON FILES
HTML FILES
SQL FILES
PICKLE FILES
DECOMPRESS
COMPRESS
Real Python