jupyter DATA INPUT & OUTPUT USING PANDAS Last Checkpoint: 5 minutes ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Markdown

Reading and Writing CSV, EXCEL, HTML & SQL FILES

```
- To allow Pandas to interact with the different file types mentioned above
- From Command Line, pip or conda install the following packages
                     sqlalchemy
                     lxml
                     html5lib
                     BeautifulSoup4
- Restart Jupyter Notebook after installing packages
- Might need to install xlrd for Excel Workbooks if necessary
- Make sure that the files are located in the same directory as your ipynb files
```

In [6]:
```
import pandas as pd
```

CSV FILES

In [9]:
```
# READ A CSV FILE

df = pd.read_csv('example')

# CALL df

df
```

Out[9]:

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
2	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
3	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63

In [11]:
```
# WRITING A CSV FILE, INDEX = FALSE WHEN YOU DON'T WANT TO IMPORT THE INDEX TO NEW FILE

df.to_csv('New File', index = False)
```

EXCEL FILES

-PANDAS CAN ALSO READ AND WRITE EXCEL FILES, HOWEVER CAN ONLY IMPORT DATA, NOT FORMULAS.

-THE EXCEL FILE MUST NOT INCLUDE IMAGES AND MACROS, OR THE READ_EXCEL METHOD MIGHT CRASH.

In [19]:
```
# READ AN EXCEL FILE

pd.read_excel('Excel_Example.xls', sheet_name = 'Sheet 1 - Table 1')
```

Out[19]:

	a	b	c	d	e	f	g	h	i	j	k
0	0	1	2	3	4	5	6	7	8	9	10
1	11	12	13	14	15	16	17	18	19	20	21
2	22	23	24	25	26	27	28	29	30	31	32
3	33	34	35	36	37	38	39	40	41	42	43

In [22]:
```
# WRITE AN EXCEL FILE

df.to_excel('Excel_Example.xls',sheet_name = 'New Sheet')
```

In [23]:
```
# LET'S TAKE A LOOK AT THIS NEW SHEET

pd.read_excel('Excel_Example.xls', sheet_name= 'New Sheet')

# NOTE THAT WE ARE TAKING DATA FROM df AND WRITING IT TO A 'NEW SHEET' IN EXCEL_EXAMPLE.XLS
```

Out[23]:

	Unnamed: 0	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
0	0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
2	2	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
3	3	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63

HTML FILES

Pandas read_html function will read tables off of a webpage and return a list of DataFrame objects

In [34]:
```
# THE FOLLOWING RETURNS A LIST OF TABLES FROM "FDIC'S FAILED BANK LIST" HTML PAGE

data = pd.read_html('https://www.fdic.gov/bank/individual/failed/banklist.html')
```

In [35]:
```
# LET'S SEE data's TYPE
type(data)
```

Out[35]:
```
list
```

In [36]:
```
# LET'S RETURN THE MAIN TABLE FROM THE HTML PAGE (HAPPENS TO BE TABLE INDEX 0)
```

In [37]:
```
data[0]
```

Out[37]:

	Bank Name	City	ST	CERT	Acquiring Institution	Closing Date
0	City National Bank of New Jersey	Newark	NJ	21111	Industrial Bank	November 1, 2019
1	Resolute Bank	Maumee	OH	58317	Buckeye State Bank	October 25, 2019
2	Louisa Community Bank	Louisa	KY	58112	Kentucky Farmers Bank Corporation	October 25, 2019
3	The Enloe State Bank	Cooper	TX	10716	Legend Bank, N. A.	May 31, 2019
4	Washington Federal Bank for Savings	Chicago	IL	30570	Royal Savings Bank	December 15, 2017
5	The Farmers and Merchants State Bank of Argonia	Argonia	KS	17719	Conway Bank	October 13, 2017
6	Fayette County Bank	Saint Elmo	IL	1802	United Fidelity Bank, fsb	May 26, 2017
7	Guaranty Bank, (d/b/a BestBank in Georgia & Mi...	Milwaukee	WI	30003	First-Citizens Bank & Trust Company	May 5, 2017

In [38]:
```
# IF YOU JUST WANT TO SEE A COUPLE OF ROWS OF DATA

data[0].head()
```

Out[38]:

	Bank Name	City	ST	CERT	Acquiring Institution	Closing Date
0	City National Bank of New Jersey	Newark	NJ	21111	Industrial Bank	November 1, 2019
1	Resolute Bank	Maumee	OH	58317	Buckeye State Bank	October 25, 2019
2	Louisa Community Bank	Louisa	KY	58112	Kentucky Farmers Bank Corporation	October 25, 2019
3	The Enloe State Bank	Cooper	TX	10716	Legend Bank, N. A.	May 31, 2019
4	Washington Federal Bank for Savings	Chicago	IL	30570	Royal Savings Bank	December 15, 2017

BONUS: CREATING AN IN-MEMORY SQLITE ENGINE

Pandas is not the best equipped to read or write to SQL. Remember to install SQLAlchemy to reduce dependency on Database-specific APIs. If you are using PostgreSQL,then psycopg2 will be a better Python package to read/write SQL files. If using MySQL, then pymysql is the package you need. However, Python has a standard library already installed for SQLite.

In this exercise, we will create a temporary SQLITE Engine Database store In-Memory in order to help Pandas read or write sql files.

The key functions for sql are:

read_sql_table(table_name, con[, schema, ...])

Read SQL database table into a DataFrame.

read_sql_query(sql, con[, index_col, ...])

Read SQL query into a DataFrame.

read_sql(sql, con[, index_col, ...])

Read SQL query or database table into a DataFrame.

DataFrame.to_sql(name, con[, flavor, ...])

Write records stored in a DataFrame to a SQL database.

In [39]:
```
# LET'S IMPORT THE CREATE_ENGINE METHOD

from sqlalchemy import create_engine
```

In [40]:
```
# LET'S CREATE THE TEMPORARY SQLITE ENGINE DATABASE AND STORE IN MEMORY

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

In [41]:
```
# LET's WRITE TO THAT TEMPORARY DATABASE

df.to_sql('Excel_Example.xls', engine)

# MAKE SURE TO READ THE DOC STRING FOR THIS METHOD AS IT HAS LOTS OF INFO
```

In [43]:
```
# LET'S PULL THIS NEWLY STORED TABLE OUT OF THE TEMPORARY DATABASE AND READ IT

sqldf = pd.read_sql('Excel_Example.xls', con = engine)
```

In [44]:
```
# LET'S DOUBLE CHECK WHAT IT LOOKS LIKE

sqldf
```

Out[44]:

	index	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
0	0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
2	2	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
3	3	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63

```
I hope you enjoyed this exercise. Please feel free to provide some feedback on how I can
improve on the material and its format. Also, share you favorite tricks when it comes to
Input/Output using Pandas.

Thanks!



Prepared by Gary-Gregoire Coquillo

If you want to use Data Science to enhance your business, let's connect and learn together!
```