CS506 Programming for Computing

HOS06D- Reading and Writing Data

06/06/2020 Created by Apiwat Chuaphan

3/28/2022 Revised by Cedric Huang

03/08/2023 Reviewed by Maram Elwakeel

School of Technology & Computing (STC) @ City University of Seattle (CityU)







Before You Start

- Version numbers may not match with the most current version at the time of writing. If given the
 option to choose between stable release (long-term support) or most recent, please select the
 stable release rather than the beta-testing version.
- There might be subtle discrepancies along with the steps. Please use your best judgment while going through this cookbook-style tutorial to complete each step.
- For your working directory, use your course number. This tutorial may use a different course number as an example.
- All the steps and concepts in this tutorial are from the textbook, so if you encounter problems in
 this tutorial, please try to read and compare the textbook to solve the problem. If you still can't
 solve the problem, please feel free to contact your course TA.
- Avoid copy-pasting code from the book or the GitHub repository. Instead, type out the code
 yourself. Resort to copy-pasting only when you are stuck and find that things are not working as
 expected.

Learning Outcomes

- Learn how to use Python's Pandas library to read and write CSV files.
- Learn how to read and write JSON files.

Learn how to read API data and serialize JSON file.

Resources

- Pandas Documentation: https://pandas.pydata.org/docs/user_guide/io.html?highlight=json#io-json-writer
- Data from Kaggle: https://www.kaggle.com/datasets

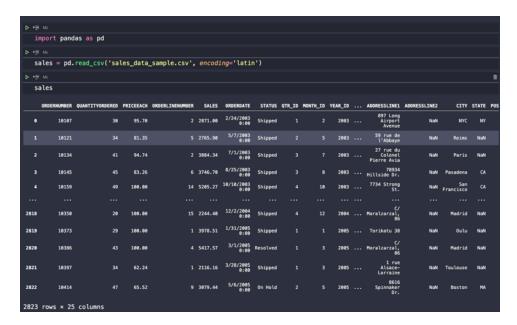
Section 1 - Read and Write

- 1) Pandas is a very powerful and popular framework for data analysis and manipulation. One of the most useful features of Pandas is the ability to read and write various types of files including CSV.
- 2) The Pandas I/O API is a set of top-level reader functions accessed like pd.read_csv() that generally return a Pandas object. More available readers and writers can be found here https://pandas.pydata.org/docs/user_guide/io.html
- 3) Common functionalities:

Format Type	Data Description	Reader	Writer
text	CSV	read_csv	to_csv
text	Fixed-Width Text File	read_fwf	
text	JSON	read_json	to_json
text	HTML	read_html	to_html
text	Local clipboard	read_clipboard	to_clipboard
	MS Excel	read_excel	to_excel

- 4) **Reading and Writing CSV** Open Jupyter Notebook:
- 5) Create a new file named sales_data_sample.csv under Module folder.
- 6) Copy a sample CSV data from https://raw.githubusercontent.com/stcta/CS612-Data-Analysis/master/example/data/sales_data_sample.csv and paste into the file we just created and then save it.

- 7) Under module folder, create a new file called reading_writing.ipynb and simply click on the file to open notebook.
- 8) Type the following into the file just created. Run selected cell to see each result.



'latin-1' or 'iso-8859-1' is the simplest text encoding maps the code points 0–255 to the bytes 0x0–0xff, which means that a string object that contains code points above U+00FF can't be encoded with this codec. Doing so will raise a **UnicodeEncodeError**

9) This data has 25 columns, you might not see the whole thing, but you still can see what they are with the columns DataFrame like below.

10) Now we will write new CSV file with multiple columns selected by the indexing operator by passing it a list of column names.

```
D P ML

data = pd.DataFrame(sales.loc[:50,['ORDERNUMBER', 'CUSTOMERNAME', 'ADDRESSLINE1', 'POSTALCODE', 'STATUS']])

data.to_csv('new_sales.csv')
```

- 11) The Series and DataFrame objects have an instance method to_csv which allows storing the contents of the object as a comma-separated-values file.
 - There are a couple common exceptions that arise when doing selections with just the indexing operator.
 - ii. If you misspell a word, you will get a KeyError
 - iii. If you forgot to use a list to contain multiple columns, you will also get a KeyError
 - iv. You just created a new CSV file named new_sales.csv with to_csv method of DataFrame.
 - v. See the file under your module folder.
- 12) Save your Jupyter Notebook with all Output