WILEY

Data Engineering

Module: Python programming

Topic: Importing and Exporting data with pandas

Scenario

- Implementing read functions available in pandas for reading different types of files and extracting relevant information from the files.
- CSV file can be loaded into a pandas dataframe using pandas read_csv().
- Excel file can be loaded into a pandas dataframe using pandas read_excel().
- JSON file can be loaded into a pandas dataframe using pandas read_json().
- Text file can be loaded into a pandas dataframe using pandas read_table()/read_fwf() depending on formatting of the file.
- XML file can be loaded into a pandas dataframe using pandas read_xml().
- HTML file can be loaded into a pandas dataframe using pandas read_html().
- SQL tables can be loaded into a pandas dataframe using pandas read_sql_table().

Background

Importing and exporting files using Pandas.

Objective

After the completing this exercise, the learner will be able to –

- Understand how to load data in python using pandas functions.
- Understand how to export pandas dataframe into different formats.

- **Problem statement**
- Load the following datasets into python using pandas.
    - births.csv
    - Golf.xlsx(Sheet1)
    - Golf.xlsx(Sheet2)
    - iris.json
    - birthsfwf.txt
    - books.xml
- Read births table from births.db(Database)
- Also load HTML table from https://en.wikipedia.org/wiki/Minnesota - (Election results from statewide races)
- Export the births dataframe into different file formats.

Dataset information

Different datasets are provided for different formats.

- Births Dataset in general consists of 4 columns which are year, month, day and births.
- Golf Dataset is about distances travelled by current golf balls and "to be designed" new golf balls.
- Iris Dataset consists of data on length and width of petals and sepals for 3 flower species. These species are setosa, virginica and versicolor.
- Books Dataset consists of data on different books. The columns are id, author, title, genre, price, publish_date and description.
- The Election Results from state wise races consists of Year, Office, GOP, DFL and others.

Download the datasets from -

https://github.com/anshupandey/WileyNXT/tree/main/DataEngineering/Importing_Exporting_data_sample_datasets

Execution

Step 1

Import the pandas library

```
1. import pandas as pd
```

Step 2

Import births.csv into python using pandas read_csv().

```
1. df=pd.read_csv("births.csv",usecols=['year','month','day','births'])
2. df.head()
```

	year	month	day	births
0	1969	1	1.0	4046
1	1969	1	1.0	4440
2	1969	1	2.0	4454
3	1969	1	2.0	4548
4	1969	1	3.0	4548

One can pass header=None in read_csv to start reading data from row 1 itself, hence with no column headers.

```
1. df1=pd.read_csv("births.csv",header=None)
2. df1.head()
```

	0	1	2	3	4
0	year	month	day	gender	births
1	1969	1	1	F	4046
2	1969	1	1	M	4440
3	1969	1	2	F	4454
4	1969	1	2	M	4548

Step 2

Import Golf.xlsx into python using pandas read_excel(). By default read_excel() reads the first sheet.

```
1. df2=pd.read_excel("Golf.xlsx")
2. df2.head()
```

	Current	New
0	264	277
1	261	269
2	267	263
3	272	266
4	258	262

To read custom sheet mention the sheet name.

```
1. df3=pd.read_excel("Golf.xlsx",sheet_name="GolfR")
2. df3.head()
```

	New	Current
0	277	264
1	269	261
2	263	267
3	266	272
4	262	258

Step 4

Import iris.json into python using pandas read_json().

```
1. df4=pd.read_json("iris.json")
2. df4.head()
```

	sepalLength	sepalWidth	petalLength	petalWidth	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

Step 5

Import table births.csv using pandas read_table().

```
1. df5=pd.read_table("births.csv",sep=",")
2. df5.head()
```

	year	month	day	gender	births
0	1969	1	1.0	F	4046
1	1969	1	1.0	M	4440
2	1969	1	2.0	F	4454
3	1969	1	2.0	M	4548
4	1969	1	3.0	F	4548

Step 6

Read a text file having fixed width formatted lines.

```
1. df6=pd.read_fwf("birthsfwf.txt",header=None)
2. df6.head()
```

	0	1	2	3	4
0	1969	1	1	F	4046
1	1969	1	1	M	4440
2	1969	1	2	F	4454
3	1969	1	2	M	4548

Step 7

Import books.xml into python using pandas read_xml().

```
1. df7=pd.read_xml("books.xml")
2. df7.head()
```

	id	author	title	genre	price	publish_date	description
0	bk101	Gambardella, Matthew	XML Developer's Guide	Computer	44.95	2000-10-01	An in-depth look at creating applications \n ...
1	bk102	Ralls, Kim	Midnight Rain	Fantasy	5.95	2000-12-16	A former architect battles corporate zombies, ...
2	bk103	Corets, Eva	Maeve Ascendant	Fantasy	5.95	2000-11-17	After the collapse of a nanotechnology \n ...
3	bk104	Corets, Eva	Oberon's Legacy	Fantasy	5.95	2001-03-10	In post-apocalypse England, the mysterious \n ...
4	bk105	Corets, Eva	The Sundered Grail	Fantasy	5.95	2001-09-10	The two daughters of Maeve, half-sisters, \n ...

Step 8

Import html table statewide election results available on https://en.wikipedia.org/wiki/Minnesota using pandas read_html().

```
1. html = pd.read_html('https://en.wikipedia.org/wiki/Minnesota',
   match='Election results from statewide races')
2. html.head()
```

```
[    Year     Office    GOP    DFL Others
 0   2020  President  45.3%  52.4%   2.3%
 1   2020    Senator  43.5%  48.8%   7.7%
 2   2018   Governor  42.4%  53.9%   3.7%
 3   2018    Senator  36.2%  60.3%   3.4%
 4   2018    Senator  42.4%  53.0%   4.6%
 5   2016  President  44.9%  46.4%   8.6%
 6   2014   Governor  44.5%  50.1%   5.4%
 7   2014    Senator  42.9%  53.2%   3.9%
 8   2012  President  45.1%  52.8%   2.1%
 9   2012    Senator  30.6%  65.3%   4.1%
 10  2010   Governor  43.2%  43.7%  13.1%
 11  2008  President  43.8%  54.1%   2.1%
 12  2008    Senator  42.0%  42.0%  16.0%
 13  2006   Governor  46.7%  45.7%   7.6%
 14  2006    Senator  37.9%  58.1%   4.0%
 15  2004  President  47.6%  51.1%   1.3%
 16  2002   Governor  44.4%  33.5%  22.1%
 17  2002    Senator  49.5%  47.3%   1.0%
 18  2000  President  45.5%  47.9%   6.6%
 19  2000    Senator  43.3%  48.8%   7.9%
 20  1998   Governor  34.3%  28.1%  37.6%
 21  1996  President  35.0%  51.1%  13.9%
 22  1996    Senator  41.3%  50.3%   8.4%
 23  1994   Governor  63.3%  34.1%   2.6%
 24  1994    Senator  49.1%  44.1%   6.8%
 25  1992  President  31.9%  43.5%  24.6%]
```

Step 9

Import Sql Table 'births' from database 'births.db'. First, create a connection to the database.

```
1. from sqlalchemy import create_engine
2.
3. # SQLAlchemy connectable
4. cnx = create_engine('sqlite:///births.db').connect()
```

Read table to dataframe.

```
1. df=pd.read_sql_table("births",columns=['year','month','day','gen
   der','births'],con=cnx)
2. df.head()
```

	year	month	day	gender	births
0	1969	1	1.0	F	4046
1	1969	1	1.0	M	4440
2	1969	1	2.0	F	4454
3	1969	1	2.0	M	4548
4	1969	1	3.0	F	4548

Step 10

Export the dataframe created in last cell into different formats.

```
1. df.to_csv("birthscsv.csv")
2. df.to_excel("birthsexcel.xlsx")
3. df.to_json("birthsjson.json")
4. df.to_xml("birthsxml.xml")
5. df.to_html("birthshtml.html")
6. df.to_sql("birthsql1",con=cnx)
```

Conclusion

We have learnt

- How to load different formats of data effectively using Pandas.
- How to export Pandas Data Frame into different formats.