


pandas III

Need help?

Technical contact to tech@argos-labs.com

May you search all operations,

- [Actions](#)
- [Verifications](#)
- [System Calls](#)
- [Interactives](#)

	<p>pandas III</p> <p>Author: Jerry Chae</p> <p><u>Now capable of processing multiple input data files. (dataframes)</u></p> <p>This is the third plugin in our pandas series. This is similar to panda-II where users can execute their Python statement sequentially like Jupyter Notebook but automatically (without Jupyter Notebook) with pandas-III. The major difference is that pandas-III enables you to take multiple data files as input.</p> <p>Primary Features</p> <p>This plugin runs python statement(s) on pandas on multiple input data files (dataframes).</p> <ul style="list-style-type: none">• https://en.wikipedia.org/wiki/Pandas_(software)• https://pandas.pydata.org/pandas-docs/stable/user_guide/10min.html <p>Prerequisite</p> <p><u>This plugin requires Python and Regular Expression skills.</u></p>
---	---

⚠ **Initial download maybe slow**

Please note that the pandas solution is a large software using numerous Python machine learning sub-modules. The bot will take more than just a few minutes to download them to be ready. **But this is just for the “first run”.** As to the second run on, the local VENV will be used to avoid downloading unless new pandas II version has been selected to replace what was in the bot originally.

Update 2021.03.11

You only need the BODY part of your pandas statements to drive the pandas-II and -III plugins.

The pandas-II and -III plugins have integrated the importing, reading, and the saving parts, you only need the body part of your statements. For example, when your pandas statements look like below you only need one line in the pandas-II and -III plugins.

For pandas-II

```
# import modules
import numpy as np
import pandas as pd
```

This pat is not needed.

```
# read data frame and save into df
df = pd.read_csv('in_file.csv')
```

```
# your scripts
df['BMI'] = df['Kilograms'] / ((df['Centimeters'] / 100.0)*(df['Centimeters'] / 100.0))
```

Only here

```
# save the result data frame
df.to_excel('out_file.xlsx',)
```

This pat is not needed.

For pandas-III

```
# import modules
import numpy as np
import pandas as pd
```

```
# read data frames and save into dfs
```

This pat is not needed.

```
dfs = list()
dfs.append(pd.read_csv('in_file1.csv')) # dfs[0]
dfs.append(pd.read_csv('in_file2.csv')) # dfs[1]
```

```
# your script
df = dfs[0].merge(dfs[1], on='sku', how='left')
```

Only here

```
# save the result data frame
df.to_excel('out_file.xlsx',)
```

This pat is not needed.

Update 2021.02.22

Sample Statements and Use of “df” and “dfs” variables

1. You must use “df” and “dfs” as variables for data-frames

- As variable for the dataframes with the Python statements in pandas II and III plugins, it is required to use "df" and "dfs" to represent dataframes (all in small cases).
- As for pandas III, the multiple dataframes ("dfs") will take [n] as index (it is zero based as the first set of dataframe becomes dfs[0]) as shown in examples below.

2. For pandas II Statements

- The "In file" will be the data frame stored at "df" Python variable
- All pandas functionality is working with "df" data frame including Reshaping at statements File
- Processed results of statement's execution will continue to be stored in the same "df" variable and eventually be the "Out file"

Plugin ▼ **pandas II** ▼ ?

Description

Plugin version 2.911.1008 ▼

Properties

In file ...

Out file ...

▼ Advanced

☐ Statements + Add Items

☒ Statements File ...

3. pandas II Statements Example

- `df['BMI'] = df['Kilograms'] / ((df['Centimeters'] / 100.0)*(df['Centimeters'] / 100.0))`
- `df = df.sort_values('BMI', ascending=False)`
- `df = df.sort_values('BMI', ascending=False).groupby('Gender').head(5)`

4. pandas III Statements

- "In files" will be a data frame stored at "dfs[0]", "dfs[1]",.... Python variable (zero base index)
- All pandas functionality is working with "dfs[n]" data frames including merge
- Processed results of statement's execution will continue to be stored in the same "df" variable and eventually be the "Out file"

Plugin ▼ **pandas III** ▼ ?

Description

Plugin version 2.915.1747 ▼

Properties

Out file ...

In files × ...
 × ...
+ Add Items

▼ Advanced

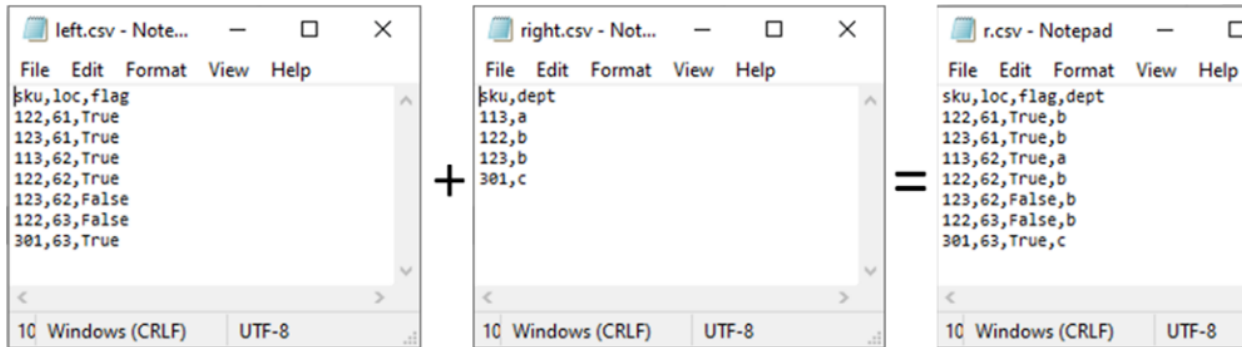
☒ Statements ×
+ Add Items

☐ Statements File ...

5. pandas III Statements Example

- `df = dfs[0].merge(dfs[1], on='sku', how='left')`

Above Python represents the process illustrated below (just like vlookup feature in Excel)



Input, Output, Features, and Parameters.

Required Input

1. Output File: One data file.

Supported input formats are **.xlsm, .xls, xlsm, .csv, .tsv, and .json**

2. Input Files: as many data files (dataframe) as you would like to process.

Supported input formats are **.xlsm, .xls, xlsm, .csv, .tsv, and .json**

Optional Input

3. Enter a **Python statement**, or multiple statements. Also a text file that contains a list of statements can be used as input.
4. When input file multiple **sheets**, you can select which sheet to be processed.
5. You can designate which row you can use as **header (variable)** for your processing.
6. You can specify a column to be used as the **index** of the dataframe.
7. You can specify which column(s) to be or not to be processed.
8. You can determine specific pandas **datatypes** for your column.
9. You can determine what character to use to **separate your data** (default is comma).
10. You can specify **encoding** technology of the input file (default is UTF-8).
11. You can select to either **show or hide the index** column in your output file.

How to set parameters

When handling multiple input data files, you must respect the input file sequence to set parameters for each one of the input files.

Properties

Out file ...

In files

1	input data file A	...	+
2	input data file B	...	
3	input data file C	...	

▼ Advanced

☐ Statements +

☐ Statements File ...

☐ Sheet Names +

☒ Header Rows

1	header row of file A	+
2	header row of file B	+

☒ Index Cols

1	index col of file A	+
2		
3	index col of file C	

☐ Use Cols +

☐ Data Types +

☐ In CSV Seps +

☐ In Encodings +

☐ Out Header

☐ Out Show Index

☐ Out CSV Sep +

☐ Out Encoding

When entering parameters for multiple input data files, you must respect the order in the [In files] section like shown in the example here. Even when there is no parameter to give to the second file (File B), **you must make one blank parameter box** to skip the 2nd data file and to set up the parameter for the 3rd data file. Omitting the following files are OK.

pandas-III plugin parameters are 100% compatible to pandas read_excel specifications

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

Please refer the parameters on the right in the pandas document above.

- Sheet Name sheet_name
- Header Row header
- Index Col index_cols
- Use Col usecols
- Data Type dtypes

Properties

Out file: output file path here ...

In files: input data file A, input data file B, input data file C

Advanced

- ☐ Statements
- ☐ Statements File
- ☐ Sheet Names
- ☒ Header Rows
 - header row of file A
 - header row of file B
- ☒ Index Cols
 - index col of file A
 - index col of file C
- ☐ Use Cols
- ☐ Data Types
- ☐ In CSV Seps: e.g. ,
- ☐ In Encodings: utf-8
- ☐ Out Header: True
- ☐ Out Show Index
- ☐ Out CSV Sep: ,
- ☐ Out Encoding: utf-8

Output file (no need to exist) in .xls .xlsx .xlsm .csv .tsv and .json

At least one or multiple input file in .xls .xlsx .xlsm .csv .tsv and .json

One Python statement or multiple python statements.

You can use a text file that contains a series of Python statements as well.

Header row default is 0 meaning the top row. You can change it here.

If index column needs to be specified, use 0 base index here.

Refer to pandas data type document.

For input files, data separation character can be changed here.

Define header to show or not for the output file.

Define whether or not the index column to show in the output file.

Define data separation character for the output file.

Define encoding standard for the output file.

All Plugins

- ABBYY Download
- ABBYY Status
- ABBYY Upload
- AD LDAP
- Adv Send Email
- API Requests
- ARGOS API
- Arithmetic Op
- ASCII Converter
- Attach Image
- AWS S3
- AWS Textra Rekog
- Base64
- Basic Numerical Operations
- Basic String Manipulation
- Bot Collabo
- Box
- Box II
- Chatwork GetMessage
- Chatwork Notification
- Citizen Log
- Clipboard
- Codat API
- Convert CharSet
- Convert Image
- Convert Image II
- Create Newfile

- [CSV2XLSX](#)
- [Data Plot I](#)
- [DeepL Free](#)
- [Detect CharSet](#)
- [Dialog Calendar](#)
- [Dialog Error](#)
- [Dialog File Selection](#)
- [Dialog Forms](#)
- [Dialog Info](#)
- [Dialog Password](#)
- [Dialog Question](#)
- [Dialog Text Entry](#)
- [Dialog Text Info](#)
- [Dialog Warning](#)
- [DirectCloud API](#)
- [Doc2TXT](#)
- [DocDigitizer Get Doc](#)
- [DocDigitizer Tracking](#)
- [DocDigitizer Upload](#)
- [Docker Remote Service](#)
- [Drag and Drop](#)
- [Dropbox](#)
- [Dynamic Python](#)
- [Email IMAP ReadMon](#)
- [Email Read Mon](#)
- [Env Check](#)
- [Env Var](#)
- [Excel2Image](#)
- [Excel Advanced](#)
- [Excel Advance IV](#)
- [Excel AdvII](#)
- [Excel AdvIII](#)
- [Excel Copy Paste](#)
- [Excel Formula](#)
- [Excel Large Files](#)
- [Excel Macro](#)
- [Excel Newfile](#)
- [Excel Simple Read](#)
- [Excel Simple Write](#)
- [Excel Style](#)
- [Excel Update](#)
- [Fairy Devices mimi AI](#)
- [File Conv](#)
- [File Downloader](#)
- [File Folder Exists](#)
- [File Folder Op](#)
- [File Status](#)
- [Fixed Form Processing](#)
- [Floating Form Processing](#)
- [Folder Monitor](#)
- [Folder Status](#)
- [Folder Structure](#)
- [FTP Server](#)
- [Git HTML Extract](#)
- [Google Calendar](#)
- [Google Cloud Vision API](#)
- [Google Drive](#)
- [Google Search API](#)
- [Google Sheets](#)
- [Google Token](#)
- [Google Translate](#)
- [Google TTS](#)
- [GraphQL API](#)
- [Html Extract](#)
- [HTML Table](#)
- [IBM Speech to Text](#)
- [IBM Visual Recognition](#)
- [Java UI Automation](#)
- [JP Holiday](#)
- [JSON Select](#)
- [JSON to from CSV](#)
- [Lazarus Forms](#)
- [Lazarus Invoices](#)
- [Lazarus RikAI](#)
- [Lazarus Riky](#)
- [LINE ID Card OCR](#)
- [LINE Notify](#)
- [LINE Receipt OCR](#)
- [Microsoft Teams](#)

- MongoDB
- MQTT Publisher
- MS Azure Text Analytics
- MS Word Extract
- NAVER OCR
- Newuser-SFDC
- OCI
- OCR PreProcess
- OpenAI API
- Oracle SQL
- Outlook
- Outlook Email
- PANDAS I
- pandas II
- pandas III
- PANDAS profiling
- Parsehub
- Password Generate
- Path Manipulation
- PDF2Doc
- PDF2Table
- PDF2TXT
- PDF Miner
- PDF SplitMerge
- PostgreSQL
- PowerShell
- PPTX Template
- Print 2 Image
- Python Selenium
- QR Generate
- QR Read
- RakurakuHanbai API
- Regression
- Rename File
- REST API
- Rossum
- Running GAS
- Scrapy Basic
- Screen Capture
- Screen Recording START
- Screen Recording STOP
- Screen Snipping
- Seaborn Plot
- SharePoint
- Simple Counter
- Simple SFDC
- Slack
- Sort CSV
- Speed Test
- SQL
- SQLite
- SSH Command
- SSH Copy
- String Manipulation
- String Similarity
- Svc Check
- Sys Info
- Telegram
- Tesseract
- Text2PDF
- Text2Word
- Text Read
- Text Write
- Time Diff
- Time Stamp
- Web Extract
- Windows Op
- Windows Screen Lock
- Win UI Control
- Win UI Text
- Word2PDF
- Word2TXT
- Word Editor
- Work Calendar
- XML Extract
- XML Manipulation
- Xtracta Get Doc
- Xtracta Tracking
- Xtracta Upload

- YouTube Operation
- ZipUnzip