learn-co-curriculum / dsc-importing-data-using-pandas-lab Public

View license

0 stars 215 forks

Star Watch

Code Issues Pull requests 1 Actions Projects Security Insights

solution ···

This branch is 13 commits ahead, 19 commits behind master. Contribute

Cheffrey2000 fixed typo ... on Sep 17, 2020 **15**

View code

README.md

Importing Data Using Pandas - Lab

Introduction

In this lab, you'll get some practice with loading files with summary or metadata, and if you find that easy, the optional "level up" content covers loading data from a corrupted csv file!

Objectives

You will be able to:

- Use pandas to import data from a CSV and and an Excel spreadsheet

Loading Files with Summary or Meta Data

Load either of the files `'Zipcode_Demos.csv'` or `'Zipcode_Demos.xlsx'`. What's going on with this dataset? Clean it up into a useable format and describe the nuances of how the data is currently formatted.

All data files are stored in a folder titled `'Data'`.

```
# Import pandas using the standard alias
import pandas as pd
```

```
# Import the file and print the first 5 rows
df = pd.read_csv('Data/Zipcode_Demos.csv')
df.head()
```

<style scoped> .dataframe tbody tr th:only-of-type { vertical-align: middle; }

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

</style>

	0	Average Statistics	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Ur
0	1	NaN	0	NaN	NaN	NaN	Na
1	2	JURISDICTION NAME	10005.8	NaN	NaN	NaN	Na
2	3	COUNT PARTICIPANTS	9.4	NaN	NaN	NaN	Na
3	4	COUNT FEMALE	4.8	NaN	NaN	NaN	Na
4	5	PERCENT FEMALE	0.404	NaN	NaN	NaN	Na

5 rows × 47 columns

```
# Print the last 5 rows of df
df.tail()
```

<style scoped> .dataframe tbody tr th:only-of-type { vertical-align: middle; }

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

</style>

	0	Average Statistics	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unna
52	53	10006	6	2	0.33	4	0.67
53	54	10007	1	0	0	1	1
54	55	10009	2	0	0	2	1
55	56	10010	0	0	0	0	0
56	57	10011	3	2	0.67	1	0.33

5 rows × 47 columns

```
# Comment: Dataframe is really two table views, one on top of the other.
# The first is a summary view of the raw data below.
# There is also a blank row at row 1 in the file.
```

```
# Clean up the dataset
prev_count = 10**3
for row in df.index:
    count = 0
    for entry in df.iloc[row].isnull():
        if entry:
            count += 1
    if count != prev_count and row!=0:
        print('On row {} there are {} null values. The previous row had {} null valu
    prev_count = count
```

```
On row 1 there are 44 null values. The previous row had 45 null values.
On row 46 there are 0 null values. The previous row had 44 null values.
```

```
# Import the first part of the data
df1 = pd.read_csv('Data/Zipcode_Demos.csv', skiprows=[1], nrows=45, usecols=[0, 1, 2
df1.head()
```

<style scoped> .dataframe tbody tr th:only-of-type { vertical-align: middle; }

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

</style>

	0	Average Statistics	Unnamed: 2
0	2	JURISDICTION NAME	10005.800
1	3	COUNT PARTICIPANTS	9.400
2	4	COUNT FEMALE	4.800
3	5	PERCENT FEMALE	0.404
4	6	COUNT MALE	4.600

```
# Look at the last five rows
df1.tail()
```

<style scoped> .dataframe tbody tr th:only-of-type { vertical-align: middle; }

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

</style>

	0	Average Statistics	Unnamed: 2
40	42	COUNT NRECEIVES PUBLIC ASSISTANCE	7.100
41	43	PERCENT NRECEIVES PUBLIC ASSISTANCE	0.649
42	44	COUNT PUBLIC ASSISTANCE UNKNOWN	0.000
43	45	PERCENT PUBLIC ASSISTANCE UNKNOWN	0.000
44	46	COUNT PUBLIC ASSISTANCE TOTAL	9.400

```
# Import the second part of the data
df2 = pd.read_csv('Data/Zipcode_Demos.csv', skiprows=47)
df2.head()
```

<style scoped> .dataframe tbody tr th:only-of-type { vertical-align: middle; }

```
    .dataframe tbody tr th {
        vertical-align: top;
    }

    .dataframe thead th {
        text-align: right;
    }
```

</style>

	47	JURISDICTION NAME	COUNT PARTICIPANTS	COUNT FEMALE	PERCENT FEMALE	COUNT MALE	PE N
0	48	10001	44	22	0.50	22	0.5
1	49	10002	35	19	0.54	16	0.4
2	50	10003	1	1	1.00	0	0.0
3	51	10004	0	0	0.00	0	0.0
4	52	10005	2	2	1.00	0	0.0

5 rows × 47 columns

Level Up (Optional) - Loading Corrupt CSV files

Occasionally, you encounter some really ill-formatted data. One example of this can be data that has strings containing commas in a csv file. Under the standard protocol, when this occurs, one is supposed to use quotes to differentiate between the commas denoting fields and the commas within those fields themselves. For example, we could have a table like this:

`ReviewerID,Rating,N_reviews,Review,VenueID 123456,4,137,This restaurant was pretty good, we had a great time.,98765`

Which should be saved like this if it were a csv (to avoid confusion with the commas in the Review text): `"ReviewerID","Rating","N_reviews","Review","VenueID" "123456","4","137","This restaurant was pretty good, we had a great time.","98765"`

Attempt to import the corrupt file, or at least a small preview of it. It is appropriately titled `'Yelp_Reviews_Corrupt.csv'` . Investigate some of the intricacies of skipping rows to then pass over this error and comment on what you think is going on.

```python
# Your code here
try:
    df = pd.read_csv('Data/Yelp_Reviews_Corrupt.csv')
except Exception as e:
    print(e)
```

```
Error tokenizing data. C error: Expected 10 fields in line 2331, saw 11
```

```python
# # Iteration 1
for i in range(1500,2000):
    try:
        df = pd.read_csv('Data/Yelp_Reviews_Corrupt.csv', nrows=i)
    except:
        print('First failure at: {}'.format(i))
        break
df1 = pd.read_csv('Data/Yelp_Reviews_Corrupt.csv', nrows=i-1)
print(len(df))
df1.head()
```

```
First failure at: 1962
1961
```

<style scoped> .dataframe tbody tr th:only-of-type { vertical-align: middle; }

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

</style>

	Unnamed: 0	business_id	cool	date	funny	
0	1	pomGBqfbxcqPv14c3XH-ZQ	0	2012-11-13	0.0	dDl8zu1
1	2	jtQARsP6P-LbkyjbO1qNGg	1	2014-10-23	1.0	LZp4UX
2	4	Ums3gaP2qM3W1XcA5r6SsQ	0	2014-09-05	0.0	jsDu6QE

	Unnamed: 0	business_id	cool	date	funny	
3	5	vgfcTvK81oD4r50NMjU2Ag	0	2011-02-25	0.0	pfavA0h
4	10	yFumR3CWzpfvTH2FCthvVw	0	2016-06-15	0.0	STiFMw

```
df1.tail()
```

<style scoped> .dataframe tbody tr th:only-of-type { vertical-align: middle; }

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

</style>

	Unnamed: 0	business_id	cool	
1956	4993	u8C8pRvaHXg3PgDrsUHJHQ	0	2 (

	Unnamed: 0	business_id	cool
1957	4998	-9nai28tnoylwViuJVrYEQ	0
1958	I had an awesome great time with friends.	NaN	NaN
1959	I loved the tapas and the excellent paella.	NaN	NaN
1960	I can't wait to come back soon.	0	otDVyX37h61WEbqPLEjCmQ

```python
# # Iteration 2
for i in range(0,500):
    try:
        temp = pd.read_csv('Data/Yelp_Reviews_Corrupt.csv', skiprows=1962, nrows=i,
    except:
        print('First failure at: {}'.format(i))
        break
df2 = pd.read_csv('Data/Yelp_Reviews_Corrupt.csv', skiprows=1962, nrows=i-1, names=d
print(len(df2))
df2.head()
```

```
498
```

<style scoped> .dataframe tbody tr th:only-of-type { vertical-align: middle; }

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

</style>

	Unnamed: 0	business_id	cool
0	STAY AWAY FROM THIS PLACE!!!!!!	5	sDofYImMQQmu4Le5G9zmpQ
1	3948	GAKFx4jFUtTOTpp_jDJnuA	0
2	3949	0QzCeORfF8EY34UODWRV9A	0
3	3950	tIt8zNrZ6_A3DmXiM-cnBA	0
4	3952	XD0LjNuPPwJPsTAHecUh7A	0

```python
temp = pd.read_csv('Data/Yelp_Reviews_Corrupt.csv')
print(len(temp))
temp.head()
```

```
---------------------------------------------------------------------------

ParserError                               Traceback (most recent call last)

<ipython-input-15-d6af0e7ded24> in <module>()
      1 # __SOLUTION__
----> 2 temp = pd.read_csv('Data/Yelp_Reviews_Corrupt.csv')
      3 print(len(temp))
      4 temp.head()


~/anaconda3/lib/python3.6/site-packages/pandas/io/parsers.py in
parser_f(filepath_or_buffer, sep, delimiter, header, names, index_col, usecols,
squeeze, prefix, mangle_dupe_cols, dtype, engine, converters, true_values,
false_values, skipinitialspace, skiprows, skipfooter, nrows, na_values,
keep_default_na, na_filter, verbose, skip_blank_lines, parse_dates,
infer_datetime_format, keep_date_col, date_parser, dayfirst, cache_dates,
iterator, chunksize, compression, thousands, decimal, lineterminator, quotechar,
quoting, doublequote, escapechar, comment, encoding, dialect, error_bad_lines,
warn_bad_lines, delim_whitespace, low_memory, memory_map, float_precision)
    683         )
    684 
--> 685         return _read(filepath_or_buffer, kwds)
    686 
    687     parser_f.__name__ = name


~/anaconda3/lib/python3.6/site-packages/pandas/io/parsers.py in
_read(filepath_or_buffer, kwds)
    461 
    462     try:
--> 463         data = parser.read(nrows)
    464     finally:
    465         parser.close()


~/anaconda3/lib/python3.6/site-packages/pandas/io/parsers.py in read(self, nrows)
   1152     def read(self, nrows=None):
   1153         nrows = _validate_integer("nrows", nrows)
-> 1154         ret = self._engine.read(nrows)
   1155 
   1156         # May alter columns / col_dict
```

```
~/anaconda3/lib/python3.6/site-packages/pandas/io/parsers.py in read(self, nrows)
   2046     def read(self, nrows=None):
   2047         try:
-> 2048             data = self._reader.read(nrows)
   2049         except StopIteration:
   2050             if self._first_chunk:


pandas/_libs/parsers.pyx in pandas._libs.parsers.TextReader.read()


pandas/_libs/parsers.pyx in pandas._libs.parsers.TextReader._read_low_memory()


pandas/_libs/parsers.pyx in pandas._libs.parsers.TextReader._read_rows()


pandas/_libs/parsers.pyx in pandas._libs.parsers.TextReader._tokenize_rows()


pandas/_libs/parsers.pyx in pandas._libs.parsers.raise_parser_error()


ParserError: Error tokenizing data. C error: Expected 10 fields in line 2331, saw
11
```

```python
temp = pd.read_csv('Data/Yelp_Reviews_Corrupt.csv', names=df1.columns, skiprows=1)
print(len(temp))
temp.head()
```

```
4651
```

<style scoped> .dataframe tbody tr th:only-of-type { vertical-align: middle; }

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

</style>

	Unnamed: 0	business_id	cool	date	funny	
0	1	pomGBqfbxcqPv14c3XH-ZQ	0	2012-11-13	0	dDI8zu1
1	2	jtQARsP6P-LbkyjbO1qNGg	1	2014-10-23	1	LZp4UX
2	4	Ums3gaP2qM3W1XcA5r6SsQ	0	2014-09-05	0	jsDu6QE
3	5	vgfcTvK81oD4r50NMjU2Ag	0	2011-02-25	0	pfavA0h
4	10	yFumR3CWzpfvTH2FCthvVw	0	2016-06-15	0	STiFMw

```
pd.read_csv('Data/Yelp_Reviews_Corrupt.csv', skiprows=len(df1)+len(df2), names=df1.c
```

<style scoped> .dataframe tbody tr th:only-of-type { vertical-align: middle; }

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

</style>

	Unnamed: 0	business_id	cool
0	Cons:	NaN	NaN
1	- Dusty! Not sure if it's all of Vegas but I...	NaN	NaN
2	- Valet parking: kinda inconvenient when you ...	NaN	NaN
3	- Sofabed is extremely flimsy	if you have more than 2 people	insist on 2 queen beds. the sofa cushions ar...
4	Other points:	NaN	NaN
5	* Should call ahead of time to make sure your...	NaN	NaN
6	* Hotel lobby is extremely small!	NaN	NaN

	Unnamed: 0	business_id	cool
7	* In-room food service was overpriced (and fo...	NaN	NaN
8	* Don't go to the 7-11	it's shady. You can shop at the am/pm or the...	NaN
9	Overall	it was a good experience for the price we pai...	3
10	4058	WPCgtEG-bJt0cZtnM-x7yw	0
11	1624	T5R6aILLDBnHQvfejY7dgA	1
12	4938	53BSdnhzcCBfBH_6TgX63Q	0
13	2897	hOB3NHuF-iVFdEkrA-PUlg	1
14	The room was lovely	we had a loft basic package and really enjoye...	NaN
15	We took part in the spa package offered throug...	NaN	NaN

	Unnamed: 0	business_id	cool
16	I emailed the concierge ahead of time to reque...	NaN	NaN
17	I've given 4 stars instead of 5 for a couple o...	the breakfast is not included in the room rat...	and the gym is adequate but quite small.
18	2422	vwQvDIb_F7AqwCPaQhHrwg	0
19	1527	04u-szAykIdu-caSDHQaKA	0
20	4080	EDcZRvERC22Cvw1yi4-VKg	1
21	4237	ZM-ljL_Y6bR4qEYsGHws5A	0
22	4498	VRTfAP2DjvUYxRY3dw37hA	0

	Unnamed: 0	business_id	cool
23	Chi	my pedicurist	was wonderful. Super sweet and very attentive...
24	Four stars instead of five because:	NaN	NaN
25	- The ladies in reception were a bit rude	both over the phone and more so in person.	NaN
26	- My pedicure only lasted two days! A pedicure...	even 3 weeks without a single chip.	7
27	2716	4VHp2gei1bpY68ZzEZE9Bg	2
28	3426	8g3u6g7J93nIOF8owARxew	0
29	Had the traditional chicken shawarma. one of m...	NaN	NaN
...	...	...	...

	Unnamed: 0	business_id	cool
2552	Starting off with drinks	Bamburger serves beer	wine
2553	The menu is varied	offering up soup	salads
2554	We went with the Bambamburger ($11.50)	which is 2/3 of a pound of prime ground chuck	and the chicken burger ($9.95) for myself
2555	Bamburger serves up great burgers	fries and shakes at a fairly good price	although if you go a little overboard with th...
2556	If you are hunting for a real deal on a burger	Bamburger might not be what you are looking for	but if you are more on the adventurous side
2557	3225	iyyWYpWm8X-6i7kBR3JHuw	0
2558	4674	4KfDcE9iU2isFpoaKeDpgw	0

	Unnamed: 0	business_id	cool
2559	4719	P4PlzIfm4uJjNmH3wY4W1Q	0
2560	3440	nW45ez1L6U4PsYhV1BTrGQ	0
2561	I had given my husband some ideas of engagemen...	a friend of ours recommended H&F.	NaN
2562	The customer service here is great - they are ...	attentive	honest and the prices are very reasonable! My...
2563	We went back for our wedding bands and I worke...	NaN	NaN
2564	We've recommended 5 other friends to H&F - the...	NaN	NaN
2565	Go to H&F for all your jewellery needs - you w...	1	PkRFSQgSfca9Tamq7b2LdQ
2566	4812	e13SEvJud_vgeDR_doL4sQ	0

	Unnamed: 0	business_id	cool
2567	1970	9NBkIExYYz3w9O5JdzDOMA	0
2568	689	BTcY04QFiS1uh-RpkR7rAg	1
2569	4874	t0T_4MM4EUHbCzBTF11FHA	0
2570	564	5XYR6doRa5Nj1JMfSDei6A	1
2571	Highly recommend the custard cakes they are th...	NaN	NaN
2572	The rice flour cake is also really good and a ...	NaN	NaN
2573	The bean cakes are great here too! orange	almond	and a few others I have tried are all good.

	Unnamed: 0	business_id	cool
2574	Can't go wrong with Nova Era	0	kBNFdviedCPFWyR-wVaAzw
2575	3458	aLcFhMe6DDJ430zelCpd2A	0
2576	This was disappointing.	NaN	NaN
2577	First off	it was really awkward sitting on the benches ...	as people walked past us while to wait for ou...
2578	Second	when we were seated	it was so loud. It felt like we were in a hig...
2579	Finally - Food was mediocre. I was extremely d...	but it wasn't flavourful.	NaN
2580	Wasn't worth the hype	unfortunately.	1
2581	4206	WdBWhGe4Siqg3IYTc4_K4A	0

2582 rows × 10 columns

Summary

Congratulations, you now practiced your Pandas-importing skills!

Releases

No releases published

Packages

No packages published

Contributors 8

Languages

● **Jupyter Notebook** 100.0%