Access the slides and files here:

https://github.com/j-berg/bioinformatics_bootcamp

#4.2

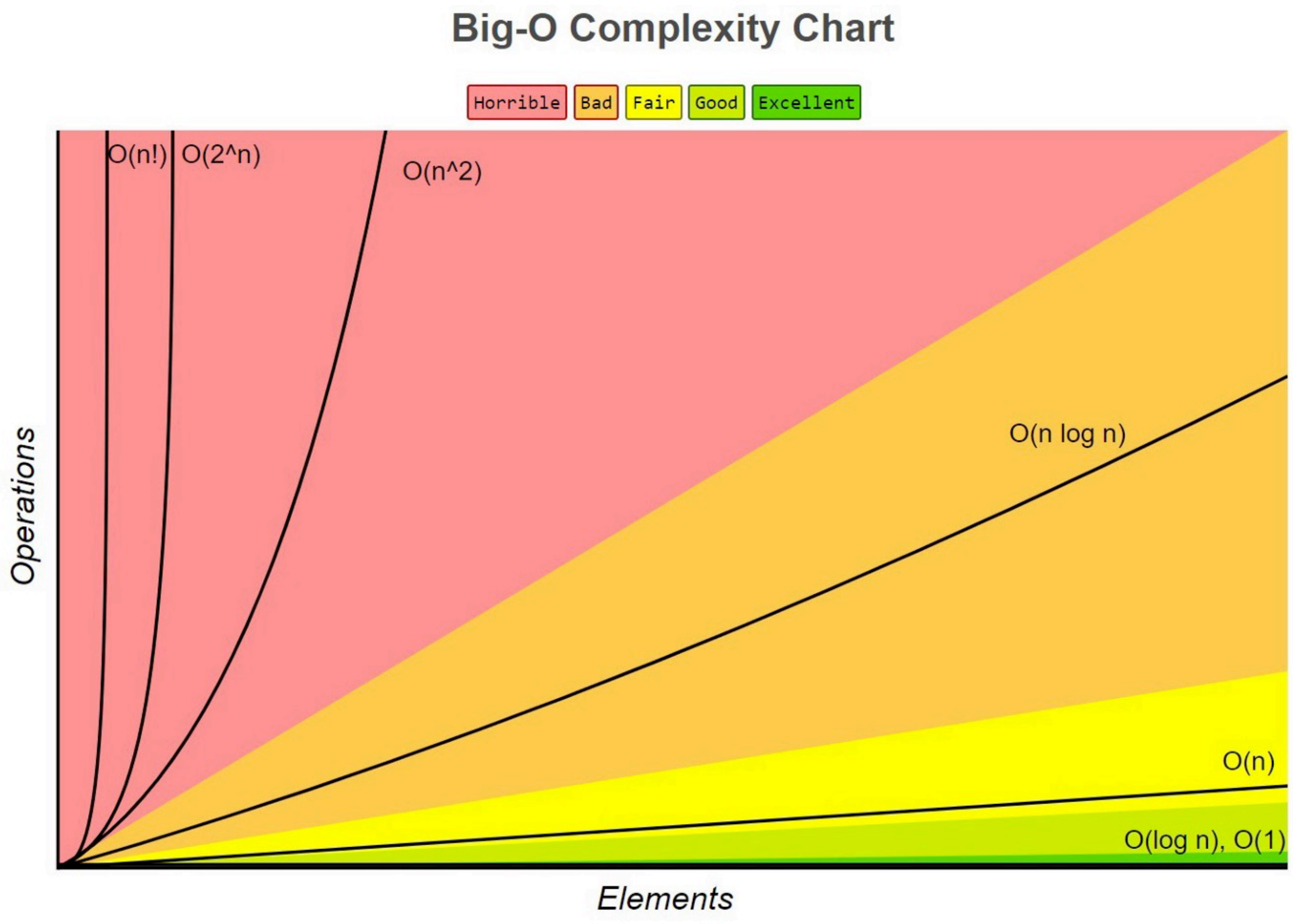
Complex tasks

Performance

Debugging

Questions from writing a script?

Performance



https://towardsdatascience.com/understanding-time-complexity-with-python-examples-2bda6e8158a7

Linear Time

```
• for x in my_list:
      if x == 100000:
            print(“I’m here”)
            break
```

Best case: item is at beginning of the list

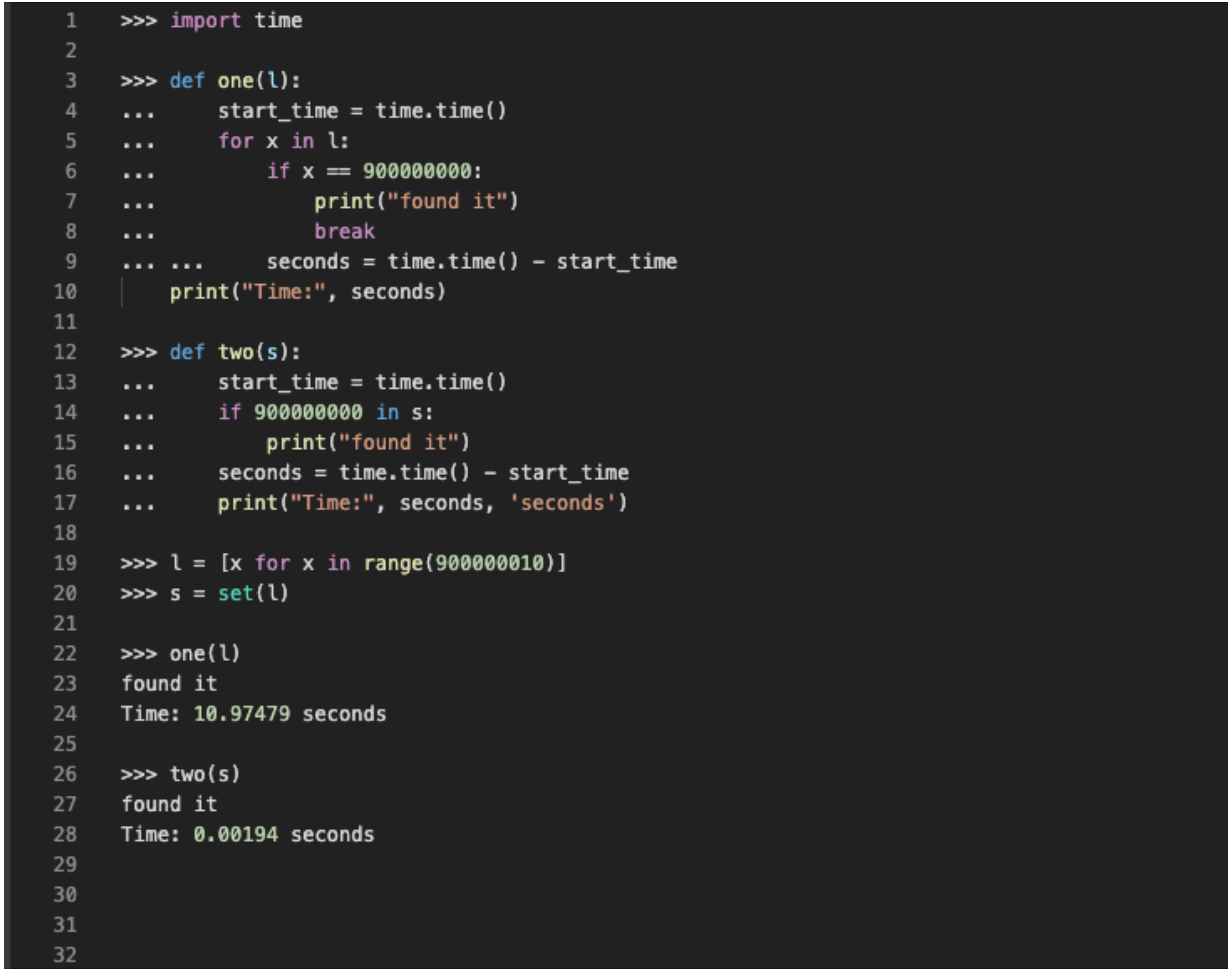
Worst case: item is at the end of the list

Constant Time

-
  ```
  if x == y:
          print(“True”)
  else:
          print(“False”)
  ```

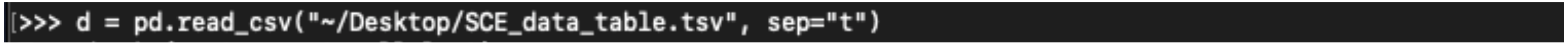
-
  ```
  if value in my_set:
          print(“True”)
  ```

- Not dependent on the input



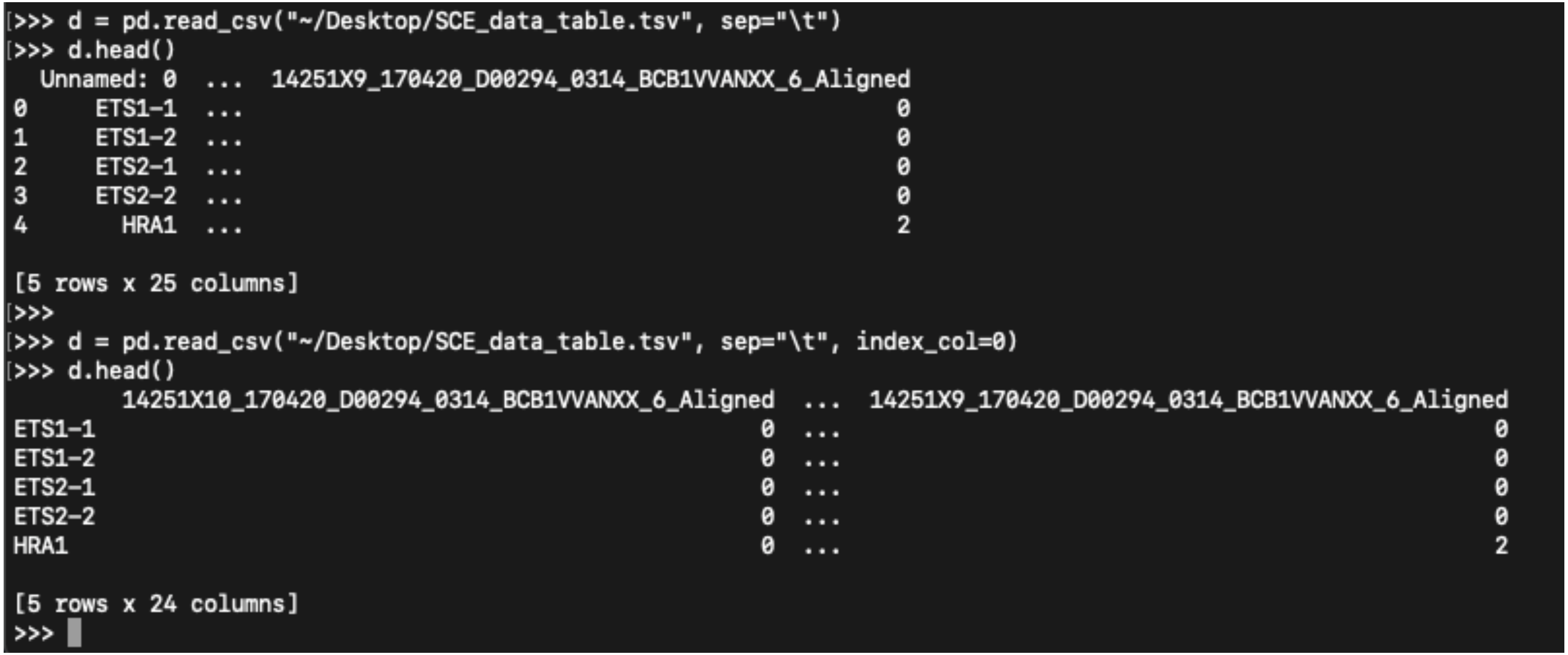
```
>>> import time

>>> def one(l):
...     start_time = time.time()
...     for x in l:
...         if x == 900000000:
...             print("found it")
...             break
... ...     seconds = time.time() - start_time
    print("Time:", seconds)

>>> def two(s):
...     start_time = time.time()
...     if 900000000 in s:
...         print("found it")
...     seconds = time.time() - start_time
...     print("Time:", seconds, 'seconds')

>>> l = [x for x in range(900000010)]
>>> s = set(l)

>>> one(l)
found it
Time: 10.97479 seconds

>>> two(s)
found it
Time: 0.00194 seconds
```

Troubleshooting



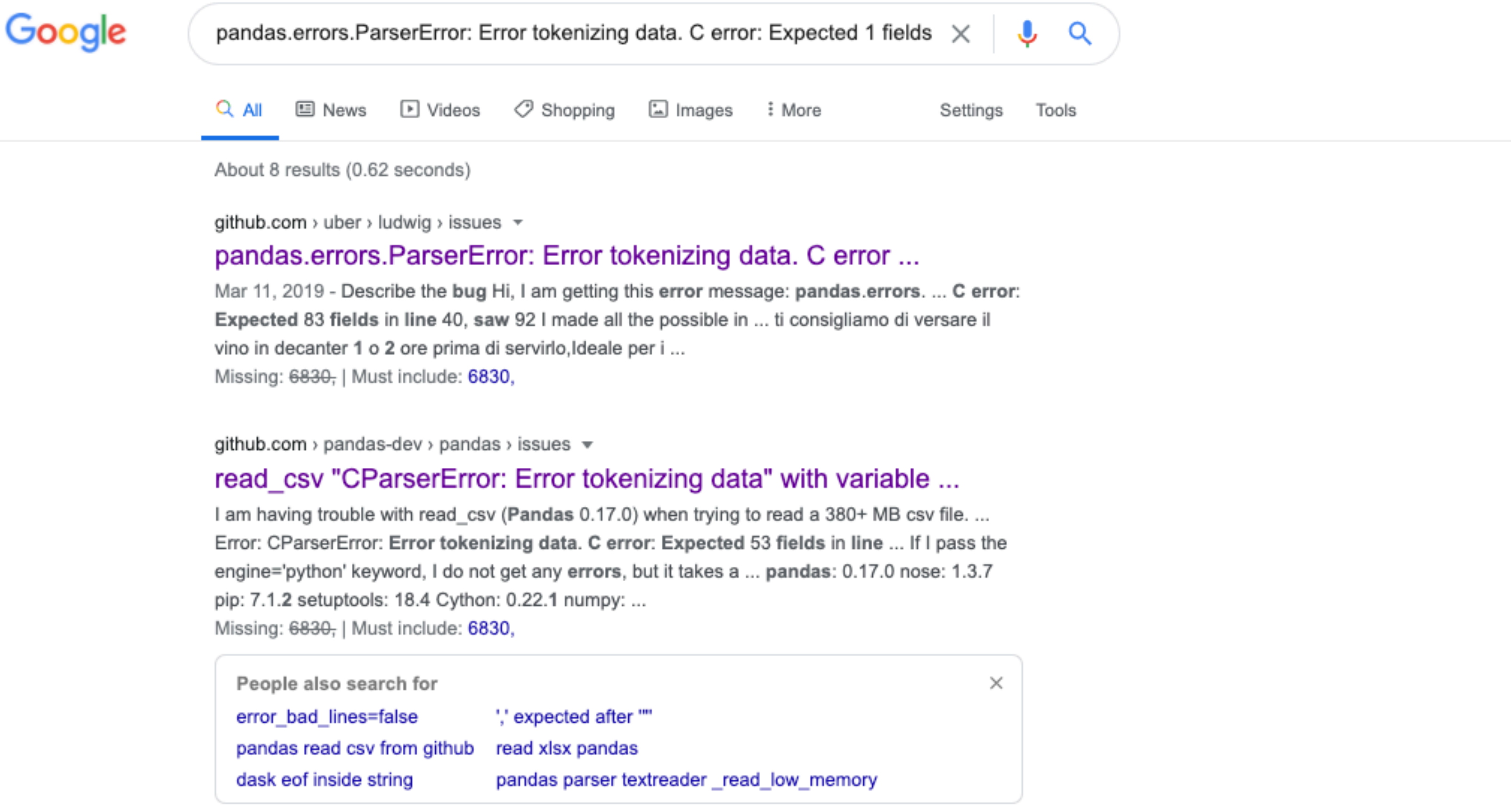
```
[>>> d = pd.read_csv("~/Desktop/SCE_data_table.tsv", sep="t")
```

Troubleshooting



```
[>>> d = pd.read_csv("~/Desktop/SCE_data_table.tsv", sep="\t")
[>>> d.head()
  Unnamed: 0  ...  14251X9_170420_D00294_0314_BCB1VVANXX_6_Aligned
0     ETS1-1  ...                                                0
1     ETS1-2  ...                                                0
2     ETS2-1  ...                                                0
3     ETS2-2  ...                                                0
4       HRA1  ...                                                2

[5 rows x 25 columns]
[>>>
[>>> d = pd.read_csv("~/Desktop/SCE_data_table.tsv", sep="\t", index_col=0)
[>>> d.head()
        14251X10_170420_D00294_0314_BCB1VVANXX_6_Aligned  ...  14251X9_170420_D00294_0314_BCB1VVANXX_6_Aligned
ETS1-1                                                 0  ...                                                0
ETS1-2                                                 0  ...                                                0
ETS2-1                                                 0  ...                                                0
ETS2-2                                                 0  ...                                                0
HRA1                                                   0  ...                                                2

[5 rows x 24 columns]
>>> 
```

Copy the most informational error line



pandas.errors.ParserError: Error tokenizing data. C error: Expected 1 fields

About 8 results (0.62 seconds)

github.com › uber › ludwig › issues

pandas.errors.ParserError: Error tokenizing data. C error ...

Mar 11, 2019 - Describe the **bug** Hi, I am getting this **error** message: **pandas.errors**. ... **C error**: **Expected** 83 **fields** in **line** 40, **saw** 92 I made all the possible in ... ti consigliamo di versare il vino in decanter **1** o **2** ore prima di servirlo,Ideale per i ...

Missing: ~~6830,~~ | Must include: 6830,

github.com › pandas-dev › pandas › issues

read_csv "CParserError: Error tokenizing data" with variable ...

I am having trouble with read_csv (**Pandas** 0.17.0) when trying to read a 380+ MB csv file. ... Error: CParserError: **Error tokenizing data**. **C error**: **Expected** 53 **fields** in **line** ... If I pass the engine='python' keyword, I do not get any **errors**, but it takes a ... **pandas**: 0.17.0 nose: 1.3.7 pip: 7.1.**2** setuptools: 18.4 Cython: 0.22.**1** numpy: ...

Missing: ~~6830,~~ | Must include: 6830,

People also search for

error_bad_lines=false

',' expected after '"'

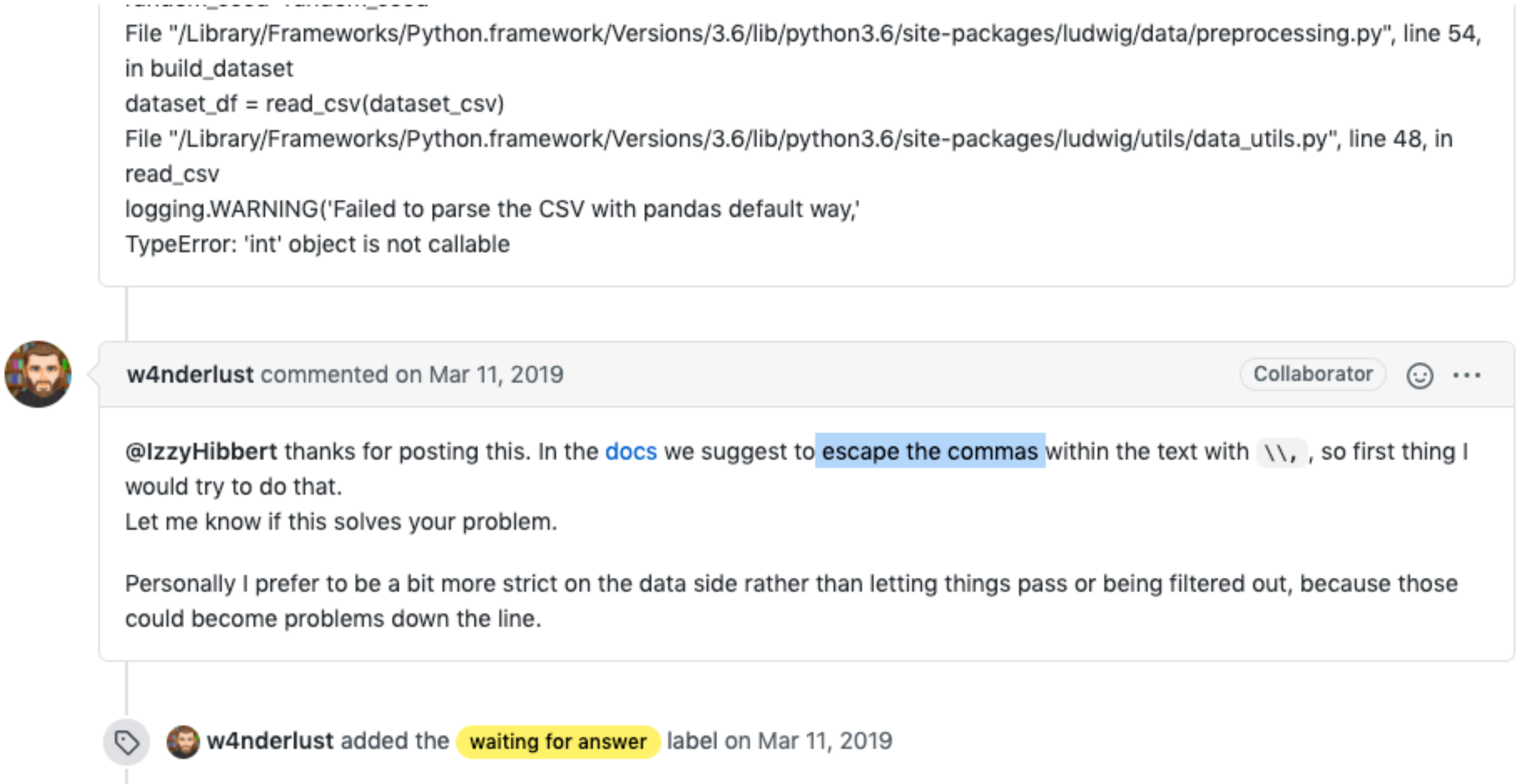
pandas read csv from github

read xlsx pandas

dask eof inside string

pandas parser textreader _read_low_memory

Not an exact answer, but we can work with it



```
File "/Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6/site-packages/ludwig/data/preprocessing.py", line 54,
in build_dataset
dataset_df = read_csv(dataset_csv)
File "/Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6/site-packages/ludwig/utils/data_utils.py", line 48, in
read_csv
logging.WARNING('Failed to parse the CSV with pandas default way,'
TypeError: 'int' object is not callable
```

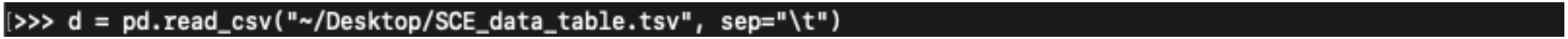
w4nderlust commented on Mar 11, 2019

@IzzyHibbert thanks for posting this. In the docs we suggest to escape the commas within the text with `\\,` , so first thing I would try to do that.
Let me know if this solves your problem.

Personally I prefer to be a bit more strict on the data side rather than letting things pass or being filtered out, because those could become problems down the line.

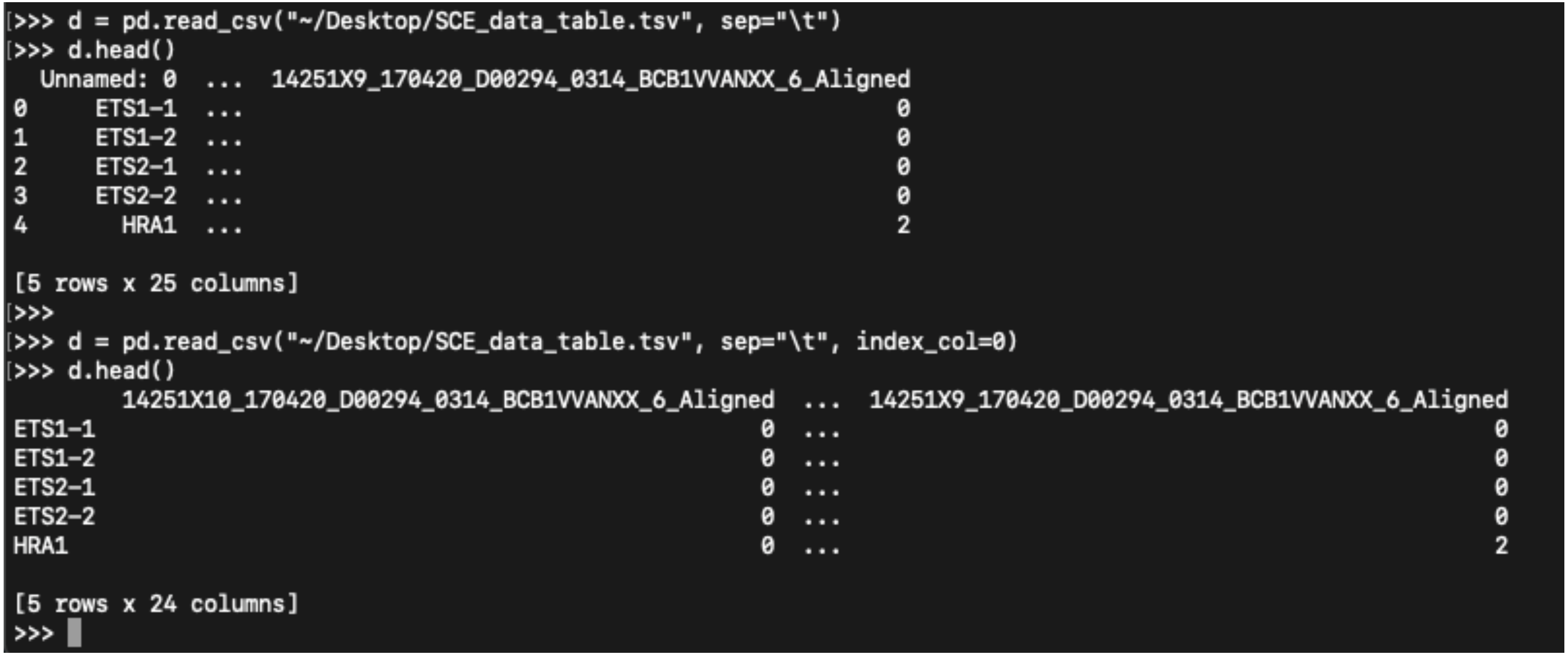
w4nderlust added the waiting for answer label on Mar 11, 2019

Troubleshooting



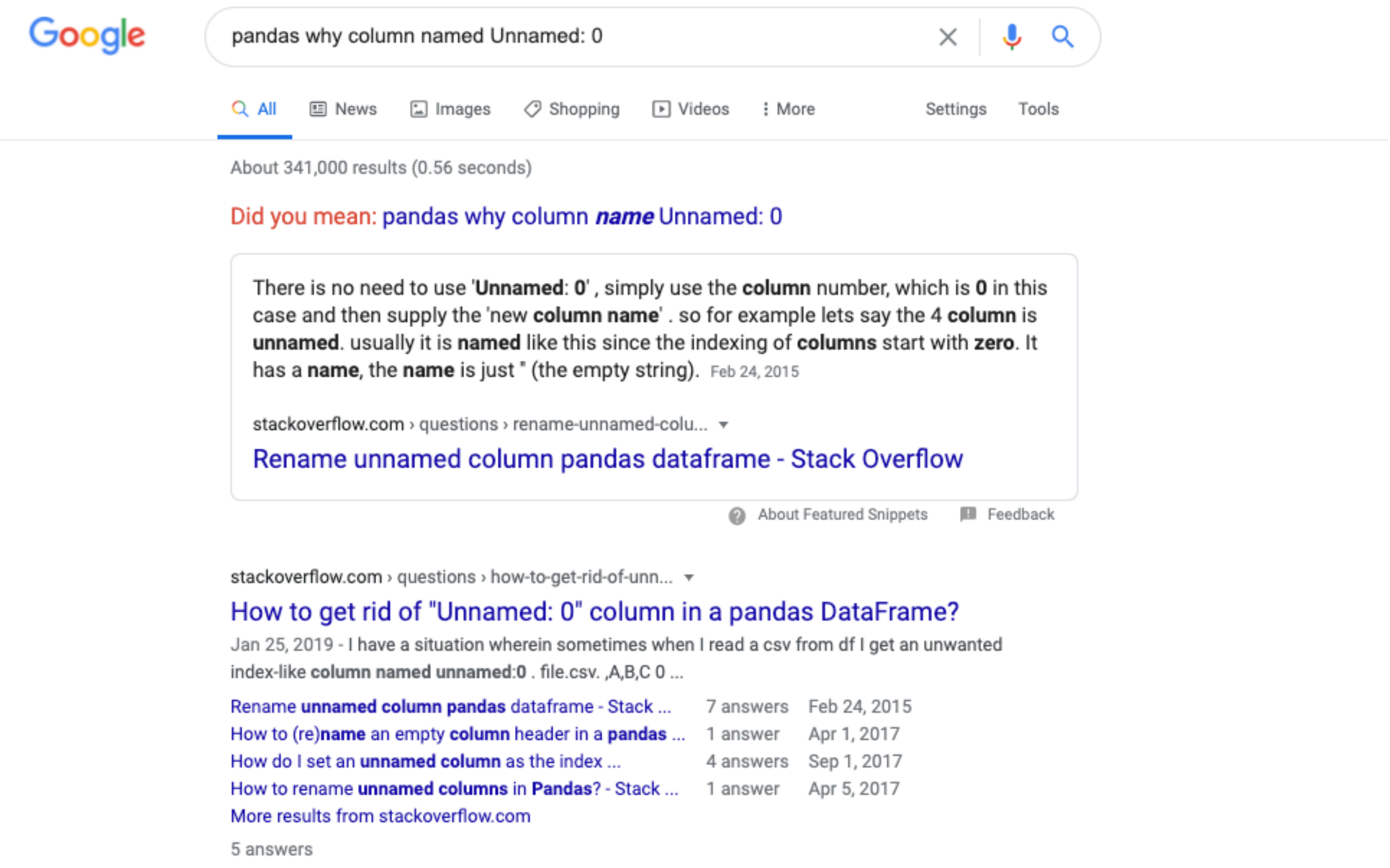
```
[>>> d = pd.read_csv("~/Desktop/SCE_data_table.tsv", sep="\t")
```

Troubleshooting



```
[>>> d = pd.read_csv("~/Desktop/SCE_data_table.tsv", sep="\t")
[>>> d.head()
  Unnamed: 0  ...  14251X9_170420_D00294_0314_BCB1VVANXX_6_Aligned
0     ETS1-1  ...                                                0
1     ETS1-2  ...                                                0
2     ETS2-1  ...                                                0
3     ETS2-2  ...                                                0
4       HRA1  ...                                                2

[5 rows x 25 columns]
[>>>
[>>> d = pd.read_csv("~/Desktop/SCE_data_table.tsv", sep="\t", index_col=0)
[>>> d.head()
        14251X10_170420_D00294_0314_BCB1VVANXX_6_Aligned  ...  14251X9_170420_D00294_0314_BCB1VVANXX_6_Aligned
ETS1-1                                                 0  ...                                                0
ETS1-2                                                 0  ...                                                0
ETS2-1                                                 0  ...                                                0
ETS2-2                                                 0  ...                                                0
HRA1                                                   0  ...                                                2

[5 rows x 24 columns]
>>> 
```

Learn to describe the problem to Google



pandas why column named Unnamed: 0

About 341,000 results (0.56 seconds)

Did you mean: pandas why column ***name*** Unnamed: 0

There is no need to use '**Unnamed**: **0**' , simply use the **column** number, which is **0** in this case and then supply the 'new **column name**' . so for example lets say the 4 **column** is **unnamed**. usually it is **named** like this since the indexing of **columns** start with **zero**. It has a **name**, the **name** is just '' (the empty string). Feb 24, 2015

stackoverflow.com › questions › rename-unnamed-colu... ▾

Rename unnamed column pandas dataframe - Stack Overflow

About Featured Snippets Feedback

stackoverflow.com › questions › how-to-get-rid-of-unn... ▾

How to get rid of "Unnamed: 0" column in a pandas DataFrame?

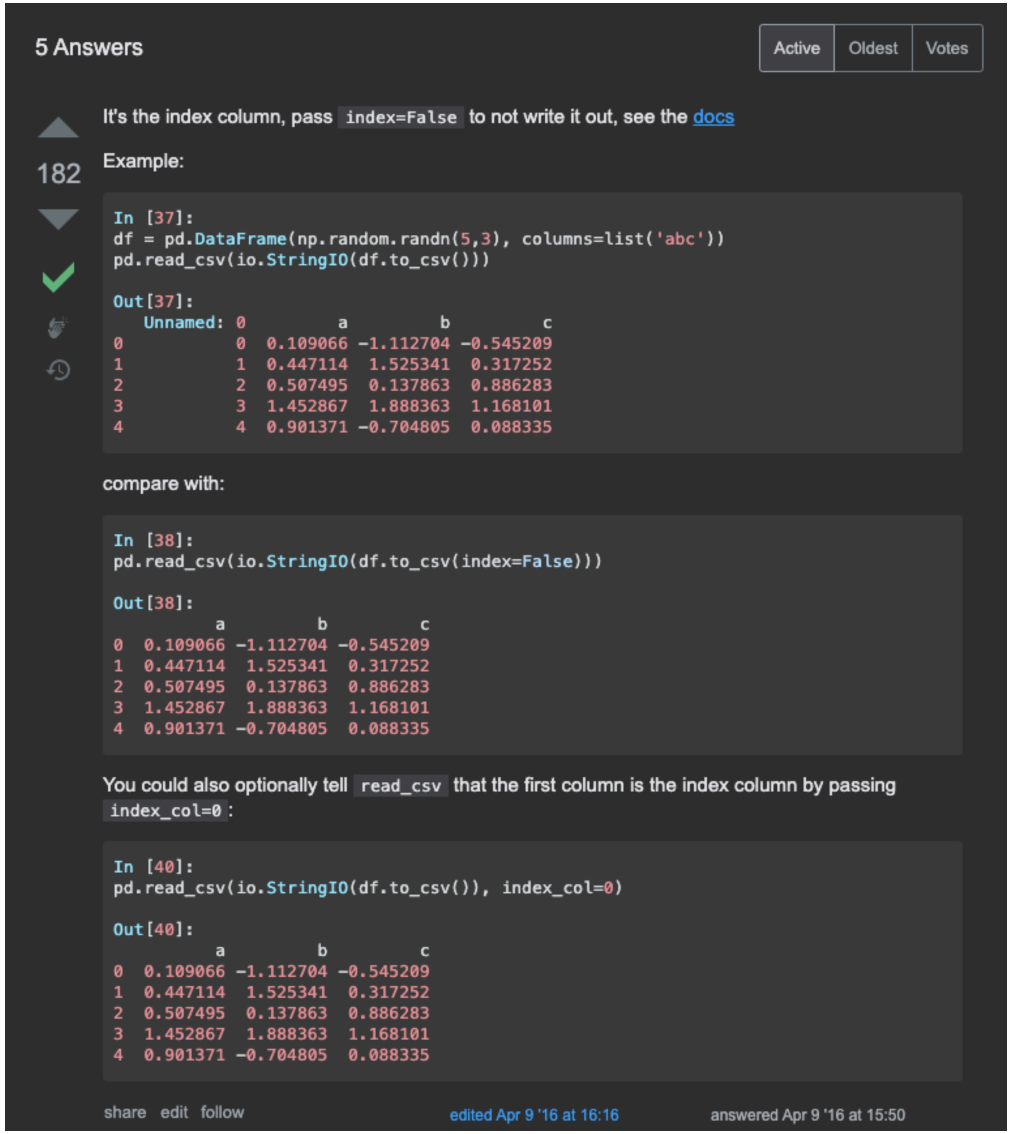
Jan 25, 2019 - I have a situation wherein sometimes when I read a csv from df I get an unwanted index-like **column named unnamed:0** . file.csv. ,A,B,C 0 ...

Rename **unnamed column pandas** dataframe - Stack ...	7 answers	Feb 24, 2015
How to (re)**name** an empty **column** header in a **pandas** ...	1 answer	Apr 1, 2017
How do I set an **unnamed column** as the index ...	4 answers	Sep 1, 2017
How to rename **unnamed columns** in **Pandas**? - Stack ...	1 answer	Apr 5, 2017

More results from stackoverflow.com

5 answers

Again, not an exact answer, but we can work with it



It's the index column, pass `index=False` to not write it out, see the docs

Example:

```
In [37]:
df = pd.DataFrame(np.random.randn(5,3), columns=list('abc'))
pd.read_csv(io.StringIO(df.to_csv()))

Out[37]:
   Unnamed: 0         a         b         c
0           0  0.109066 -1.112704 -0.545209
1           1  0.447114  1.525341  0.317252
2           2  0.507495  0.137863  0.886283
3           3  1.452867  1.888363  1.168101
4           4  0.901371 -0.704805  0.088335
```

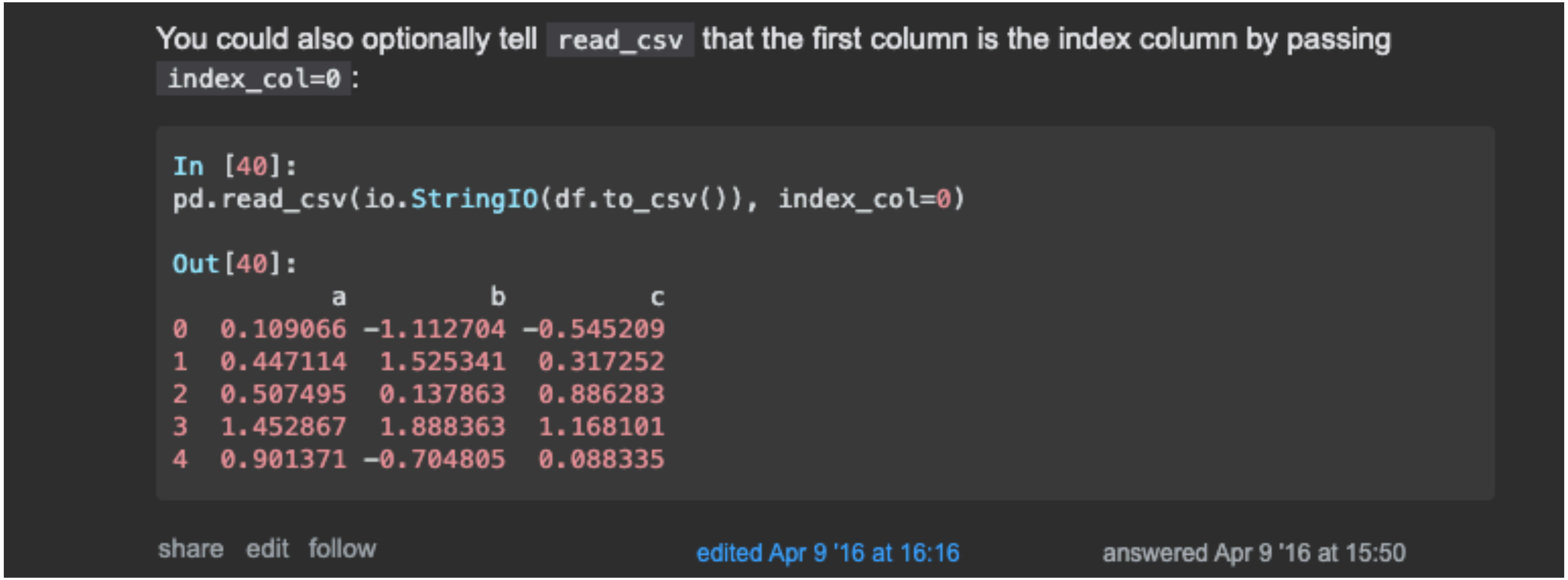
compare with:

```
In [38]:
pd.read_csv(io.StringIO(df.to_csv(index=False)))

Out[38]:
          a         b         c
0  0.109066 -1.112704 -0.545209
1  0.447114  1.525341  0.317252
2  0.507495  0.137863  0.886283
3  1.452867  1.888363  1.168101
4  0.901371 -0.704805  0.088335
```

You could also optionally tell `read_csv` that the first column is the index column by passing `index_col=0` :

```
In [40]:
pd.read_csv(io.StringIO(df.to_csv()), index_col=0)

Out[40]:
          a         b         c
0  0.109066 -1.112704 -0.545209
1  0.447114  1.525341  0.317252
2  0.507495  0.137863  0.886283
3  1.452867  1.888363  1.168101
4  0.901371 -0.704805  0.088335
```

Again, not an exact answer, but we can work with it



You could also optionally tell `read_csv` that the first column is the index column by passing `index_col=0`:

```
In [40]:
pd.read_csv(io.StringIO(df.to_csv()), index_col=0)

Out[40]:
          a         b         c
0  0.109066 -1.112704 -0.545209
1  0.447114  1.525341  0.317252
2  0.507495  0.137863  0.886283
3  1.452867  1.888363  1.168101
4  0.901371 -0.704805  0.088335
```

edited Apr 9 '16 at 16:16

answered Apr 9 '16 at 15:50

Homework

- Review concepts from Python classes
- Finish gene dictionary project