Pandas allows you to explicitly define types of the columns using dtype parameter

--

Let's say we have a large CSV file with low\_memory set to False. Where one of the columns has an integer type, but its last value is set to a random string.

Not only it takes more memory while converting the data, but the pandas also converts all the data three times (to an int, float, and string). As a result, you will get a column with an object data type.

--

Pandas can recognize it, but you need to help it a tiny bit: add the argument parse\_dates when you’reading in data from, let’s say, a comma-separated value (CSV) file:

import pandas as pd

pd.read\_csv('yourFile', parse\_dates=True)

# or this option:

pd.read\_csv('yourFile', parse\_dates=['columnName'])

There are, however, always weird date-time formats.

No worries! In such cases, you can construct your own parser to deal with this. You could, for example, make a lambda function that takes your DateTime and controls it with a format string.

import pandas as pd

dateparse~~r~~ = lambda x: pd.datetime.strptime(x, '%Y-%m-%d %H:%M:%S')

# Which makes your read command:

pd.read\_csv(infile, parse\_dates=['columnName'], date\_parser=dateparse)

# Or combine two columns into a single DateTime column

pd.read\_csv(infile, parse\_dates={'datetime': ['date', 'time']}, date\_parser=dateparse)

df = pd.DataFrame({'id' : [123,512,1857, 12354, 129, 753, 295, 610],

'colour': ['black', 'white','white','white',

'black', 'black', 'white', 'white'],

'shape': ['round', 'triangular', 'triangular','triangular','square',

'triangular','round','triangular'],

'dateArr' : ['20020127', '20030726', '20040422', '20060621', '20070319', '20100521', '20111117', '20180102']

}, columns= ['id','colour', 'shape', 'dateArr'])

DataFrame.astype({'id' : 'int32', 'colour': 'str', 'shape': 'category', 'dateArr' : 'str'})

{col: dtype, ...}

df=df.astype

df=df.astype({'id' : 'int32', 'colour': 'str', 'shape': 'category', 'dateArr' : 'str'})

df ['dateArr'] = pd.to\_datetime(df['dateArr'], format= '%Y%m%d')

filepath\_or\_buffer, sep='\t', delimiter=None, header='infer', names=None, index\_col=None, usecols=None, squeeze=False, prefix=None, mangle\_dupe\_cols=True, dtype=None, engine=None, converters=None, true\_values=None, false\_values=None, skipinitialspace=False, skiprows=None, nrows=None, na\_values=None, keep\_default\_na=True, na\_filter=True, verbose=False, skip\_blank\_lines=True, parse\_dates=False, infer\_datetime\_format=False, keep\_date\_col=False, date\_parser=None, dayfirst=False, iterator=False, chunksize=None, compression='infer', thousands=None, decimal=b'.', lineterminator=None, quotechar='"', quoting=0, escapechar=None, comment=None, encoding=None, dialect=None, tupleize\_cols=None, error\_bad\_lines=True, warn\_bad\_lines=True, skipfooter=0, skip\_footer=0, doublequote=True, delim\_whitespace=False, as\_recarray=None, compact\_ints=None, use\_unsigned=None, low\_memory=True, buffer\_lines=None, memory\_map=False, float\_precision=None)

‘Header Identification’, ‘File Title’, ‘Date Produced’, ‘Time Produced’,

Firm Data,

columns=[‘Firm Ref’, ‘Registered Firm Name’, ‘Firm Legal Status Type code’, ‘Firm Type code’, ‘Authority To Hold Client Money’, ‘Principal Address Line 1’, ‘Principal Address Line 2’, ‘Principal Address Line 3’, ‘Principal Address Line 4’, ‘Principal Address Line 5’, ‘Principal Address Line 6’, ‘Post Code Out’, ‘Post Code In’, ‘Telephone No Country Prefix’, ‘Telephone No Area Code’, ‘Telephone No Local Number’, ‘Fax Country Prefix’, ‘Fax Area Code’, ‘Fax Local Number’, ‘Current Authorisation Status code’, ‘Date Status Last Changed’, ‘Date first Authorised by Regulator’, ‘Sort Key’, ‘Last Update Date’ ]

my\_dat = pd.read\_table(r'\\NDATA13\thomai$\My Documents\Temp FCA\firms20160107.ext', nrows=10, delimiter='|')

my\_dat = pd.read\_table(r'\\NDATA13\thomai$\My Documents\Temp FCA\firms20160107.ext', nrows=10, skiprows=1)

header=None

my\_dat = pd.read\_table(r'\\NDATA13\thomai$\My Documents\Temp FCA\firms20160107.ext', nrows=10, skiprows=1, header=None, delimiter='|', names=[‘Firm Ref’, ‘Registered Firm Name’, ‘Firm Legal Status Type code’, ‘Firm Type code’, ‘Authority To Hold Client Money’, ‘Principal Address Line 1’, ‘Principal Address Line 2’, ‘Principal Address Line 3’, ‘Principal Address Line 4’, ‘Principal Address Line 5’, ‘Principal Address Line 6’, ‘Post Code Out’, ‘Post Code In’, ‘Telephone No Country Prefix’, ‘Telephone No Area Code’, ‘Telephone No Local Number’, ‘Fax Country Prefix’, ‘Fax Area Code’, ‘Fax Local Number’, ‘Current Authorisation Status code’, ‘Date Status Last Changed’, ‘Date first Authorised by Regulator’, ‘Sort Key’, ‘Last Update Date’ ])

my\_dat.columns = [' Firm Ref ',' Registered Firm Name ',' Firm Legal Status Type code ', ' Firm Type code ', ' Authority To Hold Client Money ', ' Principal Address Line 1', ' Principal Address Line 2', ' Principal Address Line 3', ' Principal Address Line 4', ' Principal Address Line 5', ' Principal Address Line 6', ' Post Code Out ', ' Post Code In ', ' Telephone No Country Prefix ', ' Telephone No Area Code ', ' Telephone No Local Number ', ' Fax Country Prefix ', ' Fax Area Code ', 'Fax Local Number', ' Current Authorisation Status code ', ' Date Status Last Changed ', ' Date first Authorised by Regulator ', ' Sort Key’', ' Last Update Date ', 'Y']

error\_bad\_lines = False

warn\_bad\_lines = True

skipfooter =1

verbose=

import pandas as pd

FCA\_field\_names = [' Firm Ref ',' Registered Firm Name ',' Firm Legal Status Type code ', ' Firm Type code ', ' Authority To Hold Client Money ', ' Principal Address Line 1', ' Principal Address Line 2', ' Principal Address Line 3', ' Principal Address Line 4', ' Principal Address Line 5', ' Principal Address Line 6', ' Post Code Out ', ' Post Code In ', ' Telephone No Country Prefix ', ' Telephone No Area Code ', ' Telephone No Local Number ', ' Fax Country Prefix ', ' Fax Area Code ', 'Fax Local Number', ' Current Authorisation Status code ', ' Date Status Last Changed ', ' Date first Authorised by Regulator ', ' Sort Key', ' Last Update Date ', 'Y']

dateparse = lambda x: pd.datetime.strptime(x, '%Y%m%d')

my\_dat = pd.read\_table(r'\\NDATA13\thomai$\My Documents\Temp FCA\firms20160107.ext', skiprows=1, header=None, delimiter='|', error\_bad\_lines = False, warn\_bad\_lines = True, skipfooter =1, names = FCA\_field\_names, parse\_dates= [' Date Status Last Changed ', ' Date first Authorised by Regulator ', ' Last Update Date '], date\_parser=dateparse )