```
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct  5 2020, 15:34:40) [MSC v.1927 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> import pandas as pd
>>> df = pd.read_csv('sbux.csv')
>>> df
            date    open    high     low   close    volume  Name
0     2013-02-08  27.920  28.325  27.920  28.185   7146296  SBUX
1     2013-02-11  28.260  28.260  27.930  28.070   5457354  SBUX
2     2013-02-12  28.000  28.275  27.975  28.130   8665592  SBUX
3     2013-02-13  28.230  28.230  27.750  27.915   7022056  SBUX
4     2013-02-14  27.765  27.905  27.675  27.775   8899188  SBUX
...          ...     ...     ...     ...     ...       ...   ...
1254  2018-02-01  56.280  56.420  55.890  56.000  14690146  SBUX
1255  2018-02-02  55.900  56.320  55.700  55.770  15358909  SBUX
1256  2018-02-05  55.530  56.260  54.570  54.690  16059955  SBUX
1257  2018-02-06  53.685  56.060  53.560  55.610  17415065  SBUX
1258  2018-02-07  55.080  55.430  54.440  54.460  13927022  SBUX

[1259 rows x 7 columns]
>>> # We want to parse the date and make a new column with the appropriate year
>>> # We will create a "date_to_year" function that will operate with each row
>>> def date_to_year(row):
        return int(row['date'].split('-')[0])

>>> df.apply(date_to_year, axis=1) # axis=1 is so that pandas applies the
date_to_year function to each ROW, not COLUMN
0       2013
1       2013
2       2013
3       2013
4       2013
        ...
1254    2018
1255    2018
1256    2018
1257    2018
1258    2018
Length: 1259, dtype: int64
>>>
```