

copy_final_answers

November 19, 2023

Extracting Stock Data Using a Web Scraping

Not all stock data is available via the API in this assignment; you will use web-scraping to obtain financial data. You will be quizzed on your results.

You will extract and share historical data from a web page using the BeautifulSoup library.

Table of Contents

1. Extracting data using BeautifulSoup
 - Download the web page Using Requests Library
 - Parse HTML on a web page using BeautifulSoup
 - Extract data and build a data frame
2. Extracting data using pandas
3. Exercise

Estimated Time Needed: 30 min

```
[4]: #!pip install pandas==1.3.3  
#!pip install requests==2.26.0  
!mamba install bs4==4.10.0 -y  
!mamba install html5lib==1.1 -y  
!pip install lxml==4.6.4  
#!pip install plotly==5.3.1
```

```
/bin/bash: mamba: command not found
```

```
/bin/bash: mamba: command not found
```

```
Collecting lxml==4.6.4
```

```
  Downloading lxml-4.6.4.tar.gz (3.2 MB)
```

```
3.2/3.2 MB 1.2 MB/s eta 0:00:00m eta
```

```
0:00:01[36m0:00:010m
```

```
  Preparing metadata (setup.py) ... done
```

```
Building wheels for collected packages: lxml
```

```
  Building wheel for lxml (setup.py) ... error
```

```
error: subprocess-exited-with-error
```

```
× python setup.py bdist_wheel did not run successfully.
```

```
  exit code: 1
```

```

> [102 lines of output]
Building lxml version 4.6.4.
/tmp/pip-install-
la1ydzxp/lxml_6c9c7dfa3c184bd188a322af0ca64ae1/setup.py:67: DeprecationWarning:
pkg_resources is deprecated as an API. See
https://setuptools.pypa.io/en/latest/pkg_resources.html
    import pkg_resources
Building without Cython.
Building against libxml2 2.10.4 and libxslt 1.1.37
Building against libxml2/libxslt in one of the following
directories:
    /home/jortizvilla/anaconda3/lib
    /home/jortizvilla/anaconda3/lib
    /home/jortizvilla/anaconda3/lib
    /home/jortizvilla/anaconda3/lib
    /home/jortizvilla/anaconda3/lib
running bdist_wheel
running build
running build_py
creating build
creating build/lib.linux-x86_64-cpython-311
creating build/lib.linux-x86_64-cpython-311/lxml
copying src/lxml/cssselect.py ->
build/lib.linux-x86_64-cpython-311/lxml
copying src/lxml/usedoctest.py ->
build/lib.linux-x86_64-cpython-311/lxml
copying src/lxml/doctestcompare.py ->
build/lib.linux-x86_64-cpython-311/lxml
copying src/lxml/ElementInclude.py ->
build/lib.linux-x86_64-cpython-311/lxml
copying src/lxml/pyclasslookup.py ->
build/lib.linux-x86_64-cpython-311/lxml
copying src/lxml/_elementpath.py ->
build/lib.linux-x86_64-cpython-311/lxml
copying src/lxml/builder.py ->
build/lib.linux-x86_64-cpython-311/lxml
copying src/lxml/__init__.py ->
build/lib.linux-x86_64-cpython-311/lxml
copying src/lxml/sax.py ->
build/lib.linux-x86_64-cpython-311/lxml
creating build/lib.linux-x86_64-cpython-311/lxml/includes
copying src/lxml/includes/__init__.py ->
build/lib.linux-x86_64-cpython-311/lxml/includes
creating build/lib.linux-x86_64-cpython-311/lxml/html
copying src/lxml/html/soupparser.py ->
build/lib.linux-x86_64-cpython-311/lxml/html
copying src/lxml/html/defs.py ->
build/lib.linux-x86_64-cpython-311/lxml/html

```

```

        copying src/lxml/html/usedoctest.py ->
build/lib.linux-x86_64-cpython-311/lxml/html
        copying src/lxml/html/_setmixin.py ->
build/lib.linux-x86_64-cpython-311/lxml/html
        copying src/lxml/html/html5parser.py ->
build/lib.linux-x86_64-cpython-311/lxml/html
        copying src/lxml/html/formfill.py ->
build/lib.linux-x86_64-cpython-311/lxml/html
        copying src/lxml/html/builder.py ->
build/lib.linux-x86_64-cpython-311/lxml/html
        copying src/lxml/html/ElementSoup.py ->
build/lib.linux-x86_64-cpython-311/lxml/html
        copying src/lxml/html/__init__.py ->
build/lib.linux-x86_64-cpython-311/lxml/html
        copying src/lxml/html/diff.py ->
build/lib.linux-x86_64-cpython-311/lxml/html
        copying src/lxml/html/clean.py ->
build/lib.linux-x86_64-cpython-311/lxml/html
        copying src/lxml/html/_html5builder.py ->
build/lib.linux-x86_64-cpython-311/lxml/html
        copying src/lxml/html/_diffcommand.py ->
build/lib.linux-x86_64-cpython-311/lxml/html
        creating build/lib.linux-x86_64-cpython-311/lxml/isoschematron
        copying src/lxml/isoschematron/__init__.py ->
build/lib.linux-x86_64-cpython-311/lxml/isoschematron
        copying src/lxml/etree.h ->
build/lib.linux-x86_64-cpython-311/lxml
        copying src/lxml/etree_api.h ->
build/lib.linux-x86_64-cpython-311/lxml
        copying src/lxml/lxml.etree.h ->
build/lib.linux-x86_64-cpython-311/lxml
        copying src/lxml/lxml.etree_api.h ->
build/lib.linux-x86_64-cpython-311/lxml
        copying src/lxml/includes/xpath.pxd ->
build/lib.linux-x86_64-cpython-311/lxml/includes
        copying src/lxml/includes/etreepublic.pxd ->
build/lib.linux-x86_64-cpython-311/lxml/includes
        copying src/lxml/includes/xslt.pxd ->
build/lib.linux-x86_64-cpython-311/lxml/includes
        copying src/lxml/includes/config.pxd ->
build/lib.linux-x86_64-cpython-311/lxml/includes
        copying src/lxml/includes/xinclude.pxd ->
build/lib.linux-x86_64-cpython-311/lxml/includes
        copying src/lxml/includes/htmlparser.pxd ->
build/lib.linux-x86_64-cpython-311/lxml/includes
        copying src/lxml/includes/relaxng.pxd ->
build/lib.linux-x86_64-cpython-311/lxml/includes
        copying src/lxml/includes/xmlparser.pxd ->

```

```

build/lib.linux-x86_64-cpython-311/lxml/includes
    copying src/lxml/includes/tree.pxd ->
build/lib.linux-x86_64-cpython-311/lxml/includes
    copying src/lxml/includes/__init__.pyd ->
build/lib.linux-x86_64-cpython-311/lxml/includes
    copying src/lxml/includes/uri.pxd ->
build/lib.linux-x86_64-cpython-311/lxml/includes
    copying src/lxml/includes/xmlerror.pxd ->
build/lib.linux-x86_64-cpython-311/lxml/includes
    copying src/lxml/includes/xmlschema.pxd ->
build/lib.linux-x86_64-cpython-311/lxml/includes
    copying src/lxml/includes/c14n.pxd ->
build/lib.linux-x86_64-cpython-311/lxml/includes
    copying src/lxml/includes/dtdvalid.pxd ->
build/lib.linux-x86_64-cpython-311/lxml/includes
    copying src/lxml/includes/schematron.pxd ->
build/lib.linux-x86_64-cpython-311/lxml/includes
    copying src/lxml/includes/lxml-version.h ->
build/lib.linux-x86_64-cpython-311/lxml/includes
    copying src/lxml/includes/etree_defs.h ->
build/lib.linux-x86_64-cpython-311/lxml/includes
    creating
build/lib.linux-x86_64-cpython-311/lxml/isoschematron/resources
    creating
build/lib.linux-x86_64-cpython-311/lxml/isoschematron/resources/rng
    copying src/lxml/isoschematron/resources/rng/iso-schematron.rng
-> build/lib.linux-x86_64-cpython-311/lxml/isoschematron/resources/rng
    creating
build/lib.linux-x86_64-cpython-311/lxml/isoschematron/resources/xsl
    copying src/lxml/isoschematron/resources/xsl/XSD2Schtrn.xsl ->
build/lib.linux-x86_64-cpython-311/lxml/isoschematron/resources/xsl
    copying src/lxml/isoschematron/resources/xsl/RNG2Schtrn.xsl ->
build/lib.linux-x86_64-cpython-311/lxml/isoschematron/resources/xsl
    creating
build/lib.linux-x86_64-cpython-311/lxml/isoschematron/resources/xsl/iso-
schematron-xslt1
    copying src/lxml/isoschematron/resources/xsl/iso-schematron-
xslt1/iso_schematron_skeleton_for_xslt1.xsl ->
build/lib.linux-x86_64-cpython-311/lxml/isoschematron/resources/xsl/iso-
schematron-xslt1
    copying src/lxml/isoschematron/resources/xsl/iso-schematron-
xslt1/iso_schematron_message.xsl ->
build/lib.linux-x86_64-cpython-311/lxml/isoschematron/resources/xsl/iso-
schematron-xslt1
    copying src/lxml/isoschematron/resources/xsl/iso-schematron-
xslt1/iso_svrl_for_xslt1.xsl ->
build/lib.linux-x86_64-cpython-311/lxml/isoschematron/resources/xsl/iso-
schematron-xslt1

```

```

    copying src/lxml/isoschematron/resources/xsl/iso-schematron-
xslt1/iso_dsd1_include.xsl ->
build/lib.linux-x86_64-cpython-311/lxml/isoschematron/resources/xsl/iso-
schematron-xslt1
    copying src/lxml/isoschematron/resources/xsl/iso-schematron-
xslt1/iso_abstract_expand.xsl ->
build/lib.linux-x86_64-cpython-311/lxml/isoschematron/resources/xsl/iso-
schematron-xslt1
    copying src/lxml/isoschematron/resources/xsl/iso-schematron-
xslt1/readme.txt ->
build/lib.linux-x86_64-cpython-311/lxml/isoschematron/resources/xsl/iso-
schematron-xslt1
    running build_ext
    building 'lxml.etree' extension
    creating build/temp.linux-x86_64-cpython-311
    creating build/temp.linux-x86_64-cpython-311/src
    creating build/temp.linux-x86_64-cpython-311/src/lxml
    gcc -pthread -B /home/jortizvilla/anaconda3/compiler_compat
-DNDEBUG -fwrapv -O2 -Wall -fPIC -O2 -isystem
/home/jortizvilla/anaconda3/include -fPIC -O2 -isystem
/home/jortizvilla/anaconda3/include -fPIC -DCYTHON_CLINE_IN_TRACEBACK=0
-I/home/jortizvilla/anaconda3/include/libxml2
-I/home/jortizvilla/anaconda3/include -Isrc -Isrc/lxml/includes
-I/home/jortizvilla/anaconda3/include/python3.11 -c src/lxml/etree.c -o
build/temp.linux-x86_64-cpython-311/src/lxml/etree.o -w
    src/lxml/etree.c:288:12: fatal error: longintrepr.h: No such file
or directory
    288 |     #include "longintrepr.h"
        |           ~~~~~
    compilation terminated.
    Compile failed: command '/usr/bin/gcc' failed with exit code 1
    creating tmp
    cc -I/home/jortizvilla/anaconda3/include/libxml2
-I/home/jortizvilla/anaconda3/include -I/usr/include/libxml2 -c
/tmp/xmlXPathInit268oysq6.c -o tmp/xmlXPathInit268oysq6.o
    /tmp/xmlXPathInit268oysq6.c: In function 'main':
    /tmp/xmlXPathInit268oysq6.c:3:5: warning: 'xmlXPathInit' is
deprecated [-Wdeprecated-declarations]
    3 |     xmlXPathInit();
        |     ~~~~~
    In file included from /tmp/xmlXPathInit268oysq6.c:1:
    /home/jortizvilla/anaconda3/include/libxml2/libxml/xpath.h:564:7:
note: declared here
    564 |     xmlXPathInit (void);
        |     ~~~~~
    cc tmp/xmlXPathInit268oysq6.o -L/home/jortizvilla/anaconda3/lib
-L/home/jortizvilla/anaconda3/lib -L/home/jortizvilla/anaconda3/lib
-L/home/jortizvilla/anaconda3/lib -L/home/jortizvilla/anaconda3/lib -lxml2 -o

```

```
a.out
error: command '/usr/bin/gcc' failed with exit code 1
[end of output]

note: This error originates from a subprocess, and is likely not a
problem with pip.
ERROR: Failed building wheel for lxml

Running setup.py clean for lxml
Failed to build lxml
ERROR: Could not build wheels for lxml, which is required to install
pyproject.toml-based projects
```

```
[17]: import pandas as pd
import requests
from bs4 import BeautifulSoup
```

In Python, you can ignore warnings using the warnings module. You can use the filterwarnings function to filter or ignore specific warning messages or categories.

```
[18]: import warnings
# Ignore all warnings
warnings.filterwarnings("ignore", category=FutureWarning)
```

0.1 Using Webscraping to Extract Stock Data Example

We will extract Netflix stock data https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0220EN-SkillsNetwork/labs/project/netflix_data_webpage.html.

In this example, we are using yahoo finance website and looking to extract Netflix data.

Fig:- Table that we need to extract

On the following web page we have a table with columns name (Date, Open, High, Low, close, adj close volume) out of which we must extract following columns

- Date
- Open
- High
- Low
- Close
- Volume

1 Steps for extracting the data

1. Send an HTTP request to the web page using the requests library.
2. Parse the HTML content of the web page using BeautifulSoup.
3. Identify the HTML tags that contain the data you want to extract.
4. Use BeautifulSoup methods to extract the data from the HTML tags.
5. Print the extracted data

1.0.1 Step 1: Send an HTTP request to the web page

You will use the request library for sending an HTTP request to the web page.

```
[19]: url = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/  
↳IBMDDeveloperSkillsNetwork-PY0220EN-SkillsNetwork/labs/project/  
↳netflix_data_webpage.html"
```

The `requests.get()` method takes a URL as its first argument, which specifies the location of the resource to be retrieved. In this case, the value of the `url` variable is passed as the argument to the `requests.get()` method, because you will store a web page URL in a `url` variable.

You use the `.text` method for extracting the HTML content as a string in order to make it readable.

```
[20]: data = requests.get(url).text  
      #print(data)
```

1.0.2 Step 2: Parse the HTML content

2 What is parsing?

In simple words, parsing refers to the process of analyzing a string of text or a data structure, usually following a set of rules or grammar, to understand its structure and meaning. Parsing involves breaking down a piece of text or data into its individual components or elements, and then analyzing those components to extract the desired information or to understand their relationships and meanings.

Next you will take the raw HTML content of a web page or a string of HTML code which needs to be parsed and transformed into a structured, hierarchical format that can be more easily analyzed and manipulated in Python. This can be done using a Python library called BeautifulSoup.

2.1 Parsing the data using the BeautifulSoup library

- Create a new BeautifulSoup object. Note: To create a BeautifulSoup object in Python, you need to pass two arguments to its constructor:
 1. The HTML or XML content that you want to parse as a string.
 2. The name of the parser that you want to use to parse the HTML or XML content. This argument is optional, and if you don't specify a parser, BeautifulSoup will use the default HTML parser included with the library. here in this lab we are using "html5lib" parser.

```
[7]: soup = BeautifulSoup(data, 'html5lib')
```

2.1.1 Step 3: Identify the HTML tags

As stated above, the web page consists of a table so, we will scrape the content of the HTML web page and convert the table into a data frame.

You will create an empty data frame using the `pd.DataFrame()` function with the following columns:
* "Date" * "Open" * "High" * "Low" * "Close" * "Volume"

```
[8]: netflix_data = pd.DataFrame(columns=["Date", "Open", "High", "Low", "Close", "Volume"])
```

Working on HTML table

These are the following tags which are used while creating HTML tables.

- `<table>`: This tag is a root tag used to define the start and end of the table. All the content of the table is enclosed within these tags.
- `<tr>`: This tag is used to define a table row. Each row of the table is defined within this tag.
- `<td>`: This tag is used to define a table cell. Each cell of the table is defined within this tag. You can specify the content of the cell between the opening and closing tags.
- `<th>`: This tag is used to define a header cell in the table. The header cell is used to describe the contents of a column or row. By default, the text inside a tag is bold and centered.
- `<tbody>`: This is the main content of the table, which is defined using the tag. It contains one or more rows of elements.

2.1.2 Step 4: Use a BeautifulSoup method for extracting data

We will use `find()` and `find_all()` methods of the BeautifulSoup object to locate the table body and table row respectively in the HTML. * The `find()` method will return particular tag content. * The `find_all()` method returns a list of all matching tags in the HTML.

```
[9]: # First we isolate the body of the table which contains all the information
# Then we loop through each row and find all the column values for each row
for row in soup.find("tbody").find_all('tr'):
    col = row.find_all("td")
    date = col[0].text
    Open = col[1].text
    high = col[2].text
    low = col[3].text
    close = col[4].text
    adj_close = col[5].text
    volume = col[6].text
```



```
# Finally we append the data of each row to the table
netflix_data = netflix_data.append({"Date":date, "Open":Open, "High":high,
↪ "Low":low, "Close":close, "Adj Close":adj_close, "Volume":volume},
↪ ignore_index=True)
```

2.1.3 Step 5: Print the extracted data

We can now print out the data frame using the head() or tail() function.

```
[10]: netflix_data.head()
```

```
[10]:
```

| | Date | Open | High | Low | Close | Volume | Adj Close |
|---|--------------|--------|--------|--------|--------|-------------|-----------|
| 0 | Jun 01, 2021 | 504.01 | 536.13 | 482.14 | 528.21 | 78,560,600 | 528.21 |
| 1 | May 01, 2021 | 512.65 | 518.95 | 478.54 | 502.81 | 66,927,600 | 502.81 |
| 2 | Apr 01, 2021 | 529.93 | 563.56 | 499.00 | 513.47 | 111,573,300 | 513.47 |
| 3 | Mar 01, 2021 | 545.57 | 556.99 | 492.85 | 521.66 | 90,183,900 | 521.66 |
| 4 | Feb 01, 2021 | 536.79 | 566.65 | 518.28 | 538.85 | 61,902,300 | 538.85 |

3 Extracting data using pandas library

We can also use the pandas read_html function from the pandas library and use the URL for extracting data.

3.1 What is read_html in pandas library?

pd.read_html(url) is a function provided by the pandas library in Python that is used to extract tables from HTML web pages. It takes in a URL as input and returns a list of all the tables found on the web page.

```
[20]: read_html_pandas_data = pd.read_html(url)
```

Or you can convert the BeautifulSoup object to a string.

```
[12]: read_html_pandas_data = pd.read_html(str(soup))
```

Because there is only one table on the page, just take the first table in the returned list.

```
[13]: netflix_dataframe = read_html_pandas_data[0]

netflix_dataframe.head()
```

```
[13]:
```

| | Date | Open | High | Low | Close* | Adj Close** | Volume |
|---|--------------|--------|--------|--------|--------|-------------|-----------|
| 0 | Jun 01, 2021 | 504.01 | 536.13 | 482.14 | 528.21 | 528.21 | 78560600 |
| 1 | May 01, 2021 | 512.65 | 518.95 | 478.54 | 502.81 | 502.81 | 66927600 |
| 2 | Apr 01, 2021 | 529.93 | 563.56 | 499.00 | 513.47 | 513.47 | 111573300 |
| 3 | Mar 01, 2021 | 545.57 | 556.99 | 492.85 | 521.66 | 521.66 | 90183900 |
| 4 | Feb 01, 2021 | 536.79 | 566.65 | 518.28 | 538.85 | 538.85 | 61902300 |

4 Exercise: use webscraping to extract stock data

Use the `requests` library to download the webpage https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0220EN-SkillsNetwork/labs/project/amazon_data_webpage.html. Save the text of the response as a variable named `html_data`.

```
[21]: html_data = requests.get('https://cf-courses-data.s3.us.cloud-object-storage.
    ↪appdomain.cloud/IBMDeveloperSkillsNetwork-PY0220EN-SkillsNetwork/labs/
    ↪project/amazon_data_webpage.html').text
```

Parse the html data using `beautiful_soup`.

```
[22]: soup = BeautifulSoup(html_data, 'html5lib')
data_frame_bs = pd.read_html(str(soup))
data_frame_bs[0]
```

```
[22]:
```

| | Date \ |
|----|---|
| 0 | Jan 01, 2021 |
| 1 | Dec 01, 2020 |
| 2 | Nov 01, 2020 |
| 3 | Oct 01, 2020 |
| 4 | Sep 01, 2020 |
| .. | ... |
| 57 | Apr 01, 2016 |
| 58 | Mar 01, 2016 |
| 59 | Feb 01, 2016 |
| 60 | Jan 01, 2016 |
| 61 | *Close price adjusted for splits.**Adjusted cl... |

| | Open \ |
|----|---|
| 0 | 3270.00 |
| 1 | 3188.50 |
| 2 | 3061.74 |
| 3 | 3208.00 |
| 4 | 3489.58 |
| .. | ... |
| 57 | 590.49 |
| 58 | 556.29 |
| 59 | 578.15 |
| 60 | 656.29 |
| 61 | *Close price adjusted for splits.**Adjusted cl... |

| | High \ |
|---|---------|
| 0 | 3363.89 |
| 1 | 3350.65 |
| 2 | 3366.80 |
| 3 | 3496.24 |

| | |
|----|---|
| 4 | 3552.25 |
| .. | ... |
| 57 | 669.98 |
| 58 | 603.24 |
| 59 | 581.80 |
| 60 | 657.72 |
| 61 | *Close price adjusted for splits.**Adjusted cl... |

| | |
|----|---|
| | Low \ |
| 0 | 3086.00 |
| 1 | 3072.82 |
| 2 | 2950.12 |
| 3 | 3019.00 |
| 4 | 2871.00 |
| .. | ... |
| 57 | 585.25 |
| 58 | 538.58 |
| 59 | 474.00 |
| 60 | 547.18 |
| 61 | *Close price adjusted for splits.**Adjusted cl... |

| | |
|----|---|
| | Close* \ |
| 0 | 3206.20 |
| 1 | 3256.93 |
| 2 | 3168.04 |
| 3 | 3036.15 |
| 4 | 3148.73 |
| .. | ... |
| 57 | 659.59 |
| 58 | 593.64 |
| 59 | 552.52 |
| 60 | 587.00 |
| 61 | *Close price adjusted for splits.**Adjusted cl... |

| | |
|----|---|
| | Adj Close** \ |
| 0 | 3206.20 |
| 1 | 3256.93 |
| 2 | 3168.04 |
| 3 | 3036.15 |
| 4 | 3148.73 |
| .. | ... |
| 57 | 659.59 |
| 58 | 593.64 |
| 59 | 552.52 |
| 60 | 587.00 |
| 61 | *Close price adjusted for splits.**Adjusted cl... |

| | Volume |
|----|---|
| 0 | 71528900 |
| 1 | 77556200 |
| 2 | 90810500 |
| 3 | 116226100 |
| 4 | 115899300 |
| .. | ... |
| 57 | 78464200 |
| 58 | 94009500 |
| 59 | 124144800 |
| 60 | 130200900 |
| 61 | *Close price adjusted for splits.**Adjusted cl... |

[62 rows x 7 columns]

Question 1: What is the content of the title attribute?

```
[23]: soup2 = soup.find_all('title')
      soup2
```

```
[23]: [<title>Amazon.com, Inc. (AMZN) Stock Historical Prices & Data - Yahoo
      Finance</title>]
```

Using BeautifulSoup, extract the table with historical share prices and store it into a data frame named `amazon_data`. The data frame should have columns Date, Open, High, Low, Close, Adj Close, and Volume. Fill in each variable with the correct data from the list `col`.

```
[30]: amazon_data = pd.DataFrame(columns=["Date", "Open", "High", "Low", "Close",
      ↪ "Adj Close", "Volume"])
      #print(dir(amazon_data))
      for row in soup.find("tbody").find_all("tr"):
          col = row.find_all("td")
          date = col[0].text
          Open = col[1].text
          high = col[2].text
          low = col[3].text
          close = col[4].text
          adj_close = col[5].text
          volume = col[6].text

          amazon_data = amazon_data._append({"Date":date, "Open":Open, "High":high,
      ↪ "Low":low, "Close":close, "Adj Close":adj_close, "Volume":volume},
      ↪ ignore_index=True)
```

Print out the first five rows of the `amazon_data` data frame you created.

```
[32]: amazon_data[0:5]
```

```
[32]:
```

| | Date | Open | High | Low | Close | Adj Close | Volume |
|---|--------------|----------|----------|----------|----------|-----------|-------------|
| 0 | Jan 01, 2021 | 3,270.00 | 3,363.89 | 3,086.00 | 3,206.20 | 3,206.20 | 71,528,900 |
| 1 | Dec 01, 2020 | 3,188.50 | 3,350.65 | 3,072.82 | 3,256.93 | 3,256.93 | 77,556,200 |
| 2 | Nov 01, 2020 | 3,061.74 | 3,366.80 | 2,950.12 | 3,168.04 | 3,168.04 | 90,810,500 |
| 3 | Oct 01, 2020 | 3,208.00 | 3,496.24 | 3,019.00 | 3,036.15 | 3,036.15 | 116,226,100 |
| 4 | Sep 01, 2020 | 3,489.58 | 3,552.25 | 2,871.00 | 3,148.73 | 3,148.73 | 115,899,300 |

Question 2: What are the names of the columns in the data frame?

```
[ ]:
```

Question 3: What is the `Open` of the last row of the `amazon_data` data frame?

```
[33]: amazon_data.iloc[-1,1]
```

```
[33]: '656.29'
```

About the Authors:

Joseph Santarcangelo has a PhD in Electrical Engineering, his research focused on using machine learning, signal processing, and computer vision to determine how videos impact human cognition. Joseph has been working for IBM since he completed his PhD.

Azim Hirjani Akansha yadav

4.1 Change Log

| Date (YYYY-MM-DD) | Version | Changed By | Change Description |
|-------------------|---------|---------------|---------------------------------------|
| 02-05-2023 | 1.3 | Akansha yadav | Updated Lab content under maintenance |
| 2021-06-09 | 1.2 | Lakshmi Holla | Added URL in question 3 |
| 2020-11-10 | 1.1 | Malika Singla | Deleted the Optional part |
| 2020-08-27 | 1.0 | Malika Singla | Added lab to GitLab |

##

© IBM Corporation 2020. All rights reserved.