[3,	<pre>stock_price = pd.read_csv('stock_price.csv') companies = pd.read_csv("companiesname.csv") company_list = list(companies["name"]) symbol_list = list(companies["symbol"]) StockPrice=pd.merge(stock_price, companies, left_on='Symbol', right_on = 'symbol')</pre>
	StockPrice = round(StockPrice, 2) StockPrice = StockPrice[['Symbol', 'name','Date','Open','High','Low','Close','Volume']] Stock_Price = pd.DataFrame(StockPrice) Stock_Price.head() Symbol name
	1 MMM 3M 2022-03-22 149.45 150.84 148.50 149.94 4252000.0 2 MMM 3M 2022-03-23 149.69 149.78 147.56 147.68 2637200.0 3 MMM 3M 2022-03-24 148.12 149.23 147.40 148.98 2028100.0 4 MMM 3M 2022-03-25 149.27 151.18 148.91 150.46 2247800.0
	<pre>from sqlalchemy import create_engine conn_url = 'postgresql://postgres:pwd4APAN5400@localhost/5400 Project' engine = create_engine(conn_url) connection = engine.connect() createCmd1 = """ DROP TABLE IF EXISTS stock_price CASCADE;</pre>
	Company_name VARCHAR(50), Date Date, Open NUMERIC(10, 2), High NUMERIC(10, 2), Low NUMERIC(10, 2), Close NUMERIC(10, 2), Volume NUMERIC(15, 2), PRIMARY KEY (Symbol, Date));
ſ	<pre>connection.execute(createCmd1) <sqlalchemy.engine.cursor.legacycursorresult 0x7fab30d4a7f0="" at=""> Stock_Price['Date'] = pd.to_datetime(Stock_Price['Date'], format = '%Y-%m-%d') Stock_Price.rename(columns={'name': 'Company_name'},inplace=True) Stock_Price.columns=Stock_Price.columns.str.lower() Stock_Price.head()</sqlalchemy.engine.cursor.legacycursorresult></pre> symbol company_name date open high low close volume
[10].	0 MMM 3M 2022-03-21 148.69 149.74 147.40 148.58 4514900.0 1 MMM 3M 2022-03-22 149.45 150.84 148.50 149.94 4252000.0 2 MMM 3M 2022-03-23 149.69 149.78 147.56 147.68 2637200.0 3 MMM 3M 2022-03-24 148.12 149.23 147.40 148.98 2028100.0 4 MMM 3M 2022-03-25 149.27 151.18 148.91 150.46 2247800.0
[10]:	<pre>Stock_Price.to_sql('stock_price', con = engine, if_exists = 'append', index = False) #TEST query1 = "SELECT * FROM stock_price" result = connection.execute (query1) row = result.fetchall() stockprice = pd.DataFrame(row, columns =['symbol','company_name', 'date','open','high','low','close', olume']) stockprice</pre>
[11]:	symbol company_name date open high low close volume 0 MMM 3M 2022-03-21 148.69 149.74 147.40 148.58 4514900.00 1 MMM 3M 2022-03-22 149.45 150.84 148.50 149.94 4252000.00 2 MMM 3M 2022-03-23 149.69 149.78 147.56 147.68 2637200.00 3 MMM 3M 2022-03-24 148.12 149.23 147.40 148.98 2028100.00 4 MMM 3M 2022-03-25 149.27 151.18 148.91 150.46 2247800.00
	.
[12]: [[13]: [[13]:	<pre>stockprice.to_csv('stockprice.csv',index=False) import pandas as pd stockprice = pd.read_csv('stockprice.csv') stockprice.head() symbol company_name date open high low close volume</pre>
	0 MMM 3M 2022-03-21 148.69 149.74 147.40 148.58 4514900.0 1 MMM 3M 2022-03-22 149.45 150.84 148.50 149.94 4252000.0 2 MMM 3M 2022-03-23 149.69 149.78 147.56 147.68 2637200.0 3 MMM 3M 2022-03-24 148.12 149.23 147.40 148.98 2028100.0 4 MMM 3M 2022-03-25 149.27 151.18 148.91 150.46 2247800.0
[2]:	<pre>import json with open('news.json') as f: news = json.load(f) print(len(news))</pre>
[57]: [[64]:	<pre>from elasticsearch import Elasticsearch from elasticsearch_dsl import connections from elasticsearch.helpers import bulk from elasticsearch_dsl import Search from elasticsearch_dsl import Q from pprint import pprint from datetime import datetime</pre> client = Elasticsearch(hosts=["localhost"],
	<pre>port=9200, http_auth=('elastic', '123456'), ca_certs='http_ca.crt', use_ssl=True, verify_certs=True) from elasticsearch.helpers import bulk resp = bulk(client, news, index = 'news') import csv</pre>
	<pre>with open('stockprice.csv') as f: reader = csv.DictReader(f) resp1 = bulk(client, reader, index='stockprice') s = Search(using=client, index="news") \</pre>
	<pre>for hit in el_response['hits']['hits']: pprint(hit['_source'].to_dict()) {'author': 'Igor Bonifacic', 'company': 'Apple', 'content': 'Following years of litigation, Canadian patent monetization firm '</pre>
	'agreement with Apple. With the deal, the two companies have ' 'settled all court cases that were ongoing between them in the ' 'US, Canada and Germany related to a ser', 'publishedAt': '2022-03-22T19:42:07Z', 'source': {'id': 'engadget', 'name': 'Engadget'}, 'symbol': 'AAPL', 'title': 'Apple settles voice over LTE patent dispute with WiLAN', 'url': 'https://www.engadget.com/apple-wilan-patent-settlement-194207518.html', 'urlToImage': 'https://s.yimg.com/os/creatr-uploaded-images/2022-03/ff7dea80-aa15-11ec-a7fd-dddc791k888e'}
	<pre>{'author': 'Kris Holt', 'company': 'Apple', 'content': 'Apple\r\n'</pre>
	"on Credit Kudos' website, which was updated on March 21st, " 'notes that the company is a subsidiary of Apple. A privacy ' 'policy link now redirects to App', 'publishedAt': '2022-03-23T11:27:19Z', 'source': {'id': 'engadget', 'name': 'Engadget'}, 'symbol': 'AAPL', 'title': 'Apple buys UK-based open banking startup Credit Kudos', 'url': 'https://www.engadget.com/apple-credit-kudos-open-banking-financial-services-112719203.html', 'urlToImage': 'https://s.yimg.com/os/creatr-images/2019-09/d0123b30-d54b-11e9-bbaf-80680876186b'} {'author': 'Jon Fingas', 'sommany': 'Apple'
	<pre>'company': 'Apple', 'content': 'It took several months, but Apple Wallet can finally hold your '</pre>
	<pre>'source': {'id': 'engadget', 'name': 'Engadget'}, 'symbol': 'AAPL', 'title': 'Arizona is the first state to allow driver's licenses in Apple '</pre>
	'content': "Apple's iPhones and other devices have become increasingly "
	<pre>'source': {'id': 'engadget', 'name': 'Engadget'}, 'symbol': 'AAPL', 'title': 'Apple is reportedly planning an iPhone hardware subscription '</pre>
	<pre>'next-generation iPad Pro sometime this fall. In his latest Power ' 'On newsletter\r\n' ', Gurman says he anticipates the new tablet will feature [+934 ' 'chars]', 'description': 'Bloomberg reporter Mark Gurman expects Apple will release its ' 'next-generation iPad Pro sometime this fall. In his latest ' 'Power On newsletter\r\n' ', Gurman says he anticipates the new tablet will feature ' 'MagSafe inductive charging and Apple's long-rumored but as y', 'publishedAt': '2022-03-27T18:01:38Z', 'source': {'id': 'engadget', 'name': 'Engadget'},</pre>
	-
	-
	'source': {'id': 'engadget', 'name': 'Engadget'}, 'symbol': 'AAPL', 'title': 'Here's why your iOS auto-updates often arrive late', 'url': 'https://www.engadget.com/apple-ios-auto-update-roll-out-craig-federighi-114547796.html', 'urlToImage': 'https://s.yimg.com/os/creatr-uploaded-images/2022-03/489e27a0-b01c-llec-8f4a-cfa9c6a4e02a'} {'author': 'Jon Fingas', 'company': 'Apple', 'content': 'Apple is well-known for doing things in-house whenever possible, '
	<pre>'and that might even extend to financial services.' 'Bloombergsources claim Apple is laying the groundwork for "future ' 'financial services [+1457 chars]', 'description': 'Apple is well-known for doing things in-house whenever '</pre>
	'title': 'Apple reportedly wants to handle more financial services in-house', 'url': 'https://www.engadget.com/apple-financial-services-lending-credit-checks-194439130.html', 'urlToImage': 'https://s.yimg.com/os/creatr-images/2019-11/55a56690-0496-11ea-bd9e-d8548107e12e'} {'author': 'Steve Dent', 'company': 'Apple', 'content': "If you've been eyeing the latest Apple Watch Series 7 but are "
	<pre>'waiting for a deal, it may be time to act. No less than four '</pre>
	<pre>'urlToImage': 'https://s.yimg.com/os/creatr-uploaded-images/2022-04/b030ba80-b3ec-11ec-bbef-6937241f 649e'} {'author': 'Khamosh Pathak', 'company': 'Apple', 'content': 'Every computer develops some cruft over time: the small app you '</pre>
	<pre>'utility you downloaded when you wanted to free up some space, '</pre>
	<pre>{'author': 'Jake Peterson', 'company': 'Apple', 'content': 'Back in the 90s, you needed to have powerful hardware to use an '</pre>
	<pre>'time. In case you', 'publishedAt': '2022-04-05T21:00:00Z', 'source': {'id': None, 'name': 'Lifehacker.com'}, 'symbol': 'AAPL', 'title': "Relive the '90s With These Web-Based Mac Emulators", 'url': 'https://lifehacker.com/relive-the-90s-with-these-web-based-mac-emulators-1848752013', 'urlToImage': 'https://i.kinja-img.com/gawker-media/image/upload/c_fill,f_auto,fl_progressive,g_center,h_675,pg_1,q_80,w_1200/a1a3b20fcd9de8b462b5d06b662ff4a5.png'}</pre>
[96]:	<pre>TypeError</pre>
	<pre>.sort({'date': {'order': 'asc'}}) # by default you get 10 hits result = [entry['_source'] for entry in s['hits']['hits']] AttributeError</pre>
[94]:	<pre>3 .sort({'date': {'order': 'asc'}}) 4 # by default you get 10 hits 5 AttributeError: 'dict' object has no attribute 'query' result {'symbol': 'AAPL', 'company_name': 'Apple', 'date': '2022-04}</pre>
137]:	<pre>from flask import Flask, render_template, request from doctest import debug from datetime import date import matplotlib.pyplot as plt import io import base64</pre>
178]:	<pre>from matplotlib.backends.backend_agg import FigureCanvasAgg as FigureCanvas app = Flask(name) es = client @app.route("/", methods=['GET']) def index(): return render_template('index.html')</pre>
	<pre>@app.route('/search/', methods=['POST']) def search_stock(): if request.method == 'POST': data = request.form.to_dict(flat=False) print(data) company = None if data['company'][0] == '' else data['company'][0] start_date = None if data['start_date'][0] == '' else data['start_date'][0] end_date = None if data['end_date'][0] == '' else data['end_date'][0] print(company)</pre>
	<pre>conditions = {} conditions1 = {} query = [] query1 = [] if company is not None: if (company in company_list): query.append({"match" : {"company" : company}}) elif (company in symbol_list): query.append({"match" : {"symbol" : company}})</pre>
	<pre>else:</pre>
	<pre>if company is not None: if (company in company_list): query1.append({"match" : {"company_name" : company}}) elif (company in symbol_list): query1.append({"match" : {"symbol" : company}}) else: print("No data found") if start_date is not None: query1.append({"range": {"date": {"gte": start_date}}}) if end date is not None:</pre>
	<pre>query1.append({"range": {"date": {"lte": end_date}}}) conditions1['sort'] = {"date": {"order": "asc"}} conditions1['query'] = {"bool":{</pre>
	<pre>res_stockprice = es.search(index = "stockprice",</pre>
	<pre>yAxis1 = [float(i["volume"]) for i in result1] # Create figure and axis #1 fig, ax1 = plt.subplots() # plot line chart on axis #1 ax1.plot(xAxis,yAxis,marker="o") ax1.set_ylabel('close') ax1.set_ylim(min(yAxis)-2, max(yAxis)+2) ax1.legend(['Close Price(\$)'], loc="upper left") ax1.set_xlabel('Date')</pre>
	<pre>ax1.set_xticklabels(ax1.get_xticks(), rotation = 15) # set up the 2nd axis ax2 = ax1.twinx() # plot bar chart on axis #2 ax2.bar(xAxis,yAxis1, width=0.5, alpha=0.5, color='orange') ax2.grid(False) # turn off grid #2 ax2.set_ylabel('Volume') digit = len(str(max(yAxis1))) - 3 ax2.set_ylim(0, max(yAxis1)+10**digit) ax2.legend(['Volume'], loc="upper right")</pre>
	<pre># Convert plot to PNG image pngImage = io.BytesIO() FigureCanvas(fig).print_png(pngImage) # Encode PNG image to base64 string pngImageB64String = "data:image/png;base64," pngImageB64String += base64.b64encode(pngImage.getvalue()).decode('utf8') plt.close()</pre>
	<pre>return render_template('result.html', company = company, result=result, column_names = column ames,</pre>
	<pre>* Environment: production WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead. * Debug mode: off * Running on http://localhost:10030/ (Press CTRL+C to quit) 127.0.0.1 [25/Apr/2022 14:40:55] "GET / HTTP/1.1" 200 - /var/folders/vl/2g8y29gx603c00ljzxssjc0000gn/T/ipykernel_35052/2103257377.py:56: DeprecationWarnin g: The 'body' parameter is deprecated for the 'search' API and will be removed in a future version. Instead use API parameters directly. See https://github.com/elastic/elasticsearch-py/issues/1698 for more information</pre>
	res_news = es.search(index = "news", /var/folders/vl/2g8y29gx603c00ljzxssjc0000gn/T/ipykernel_35052/2103257377.py:60: DeprecationWarnin g: The 'body' parameter is deprecated for the 'search' API and will be removed in a future version. In nstead use API parameters directly. See https://github.com/elastic/elasticsearch-py/issues/1698 for nore information res_stockprice = es.search(index = "stockprice", /var/folders/vl/2g8y29gx603c00ljzxssjc0000gn/T/ipykernel_35052/2103257377.py:81: UserWarning: Fixed Formatter should only be used together with FixedLocator ax1.set_xticklabels(ax1.get_xticks(), rotation = 15) 127.0.0.1 - [25/Apr/2022 14:41:00] "POST /search/ HTTP/1.1" 200 -
	{'company': ['Amazon'], 'start_date': ['2022-04-03'], 'end_date': ['2022-04-17']} Amazon 127.0.0.1 [25/Apr/2022 14:52:17] "GET / HTTP/1.1" 200 - 127.0.0.1 [25/Apr/2022 14:53:23] "POST /search/ HTTP/1.1" 200 - {'company': ['Apple'], 'start_date': ['2022-04-03'], 'end_date': ['2022-04-18']} Apple
163]: 163]:	<pre>client.indices.delete(index='news', ignore=[400, 404]) client.indices.delete(index='stockprice', ignore=[400, 404]) num = 2494100.00 digit = len(str(num)) -4 10**digit +num 2594100.0 condition = {"sort":{"publishedAt": {"order": "desc"}},</pre>
,] :	<pre>"query":{ "bool":{ "must": {"match" : {"company" : "Apple"}}, "filter": [</pre>
	<pre>res_news = client.search(index = "news",</pre>
[97]:	result /var/folders/vl/2g8y29gx603c00ljzxssjc0000gn/T/ipykernel_35052/1090609150.py:12: DeprecationWarnin g: The 'body' parameter is deprecated for the 'search' API and will be removed in a future version. Instead use API parameters directly. See https://github.com/elastic/elasticsearch-py/issues/1698 for more information res_news = client.search(index = "news",
	result /var/folders/v1/2g8y29gx603c001jzxssjc0000gn/T/ipykernel_35052/1090609150.py:12: DeprecationWarnin g: The 'body' parameter is deprecated for the 'search' API and will be removed in a future version. I nstead use API parameters directly. See https://github.com/elastic/elasticsearch-py/issues/1698 for r ore information res_news = client.search(index = "news", [{'source': {'id': 'engadget', 'name': 'Engadget'}, 'author': 'Kris Holt', 'title': "Apple signs an exclusive deal with Tom Hanks' production company", 'description': "Apple\r\n is looking to build on the success of TV+ by locking in Tom Hanks'\r\n production company to an exclusive multi-year deal. The agreement with Playtone covers series, document aries and unscripted projects.Hanks is also set to star in a sequel to World Wa", 'url': 'https://www.engadget.com/apple-tv-plus-tom-hanks-playtone-deal-greyhound-163749926.html', 'urlToImage': 'https://s.yimg.com/os/creatr-uploaded-images/2022-04/da9c59e0-bb45-11ec-bbff-79cebaee84f3', 'publishedAt': '2022-04-13T16:37:492',
	/var/folders/v1/2g8y29gx603c001jzx_ssjc0000gn/T/ipykernel_35052/1090609150.py:12: DeprecationWarnin g: The 'body' parameter is deprecated for the 'search' API and will be removed in a future version. Instead use API parameters directly. See https://github.com/elastic/elasticsearch-py/issues/1698 for rore information res_news = client.search(index = "news", [{'source': ('id': 'engadget', 'name': 'Engadget'}, 'author': 'Kris Holt', 'title': "Apple signs an exclusive deal with Tom Hanks' production company", 'description': "Apple\r\n is looking to build on the success of TV+ by locking in Tom Hanks'\r\n production company to an exclusive multi-year deal. The agreement with Playtone covers series, document aries and unscripted projects.Hanks is also set to star in a sequel to World Wa", 'url': 'https://www.engadget.com/apple-tv-plus-tom-hanks-playtone-deal-greyhound-163749926.html', 'urlToImage': 'https://s.yimg.com/os/creatr-uploaded-images/2022-04/da9c59e0-bb45-1lec-bbff-79cebare84f37, 'publishedAt': '2022-04-13T16:37:49Z', 'content': "Apple\r\n is looking to build on the success of TV+ by locking in Tom Hanks'\r\n production company to an exclusive multi-year deal. The agreement with Playtone covers series, documentarics and unscripte [+1280 chars]", 'company': 'Apple', 'symbol': 'AAPL'), {'source': {'id': None, 'name': 'Lifehacker.com'}, 'author': 'Khamosh Pathak', 'title': 'The Fastest Ways to Convert Your iPhone HEIC Photos to JPG (So You Can Actually Use The mol', 'description': 'HEIC is a solid format for Apple devices. It includes Live Photos, Apple's post-pro-
	/var/folders/v1/2g8y29gx603c001jzx_ssjc0000gn/T/ipykernel_35052/1090609150.py:12: DeprecationWarning: The 'body' parameter is deprecated for the 'search' API and will be removed in a future version. Instead use API parameters directly. See https://github.com/elastic/elasticsearch-py/issues/1698 for rore information res_news = client.search(index = "news", [{'source': {'id': 'engadget', 'name': 'Engadget'},
	/war/folders/v1/2g8y2gx633c001/zx_ssjc000cgn/T/ipykernel_35052/1090609150.py:12: DeprecationWarmin gr. The 'body' parameter is deprecated for the 'search' API and will be removed in a future version.' satead use API parameters directly. See https://github.com/elastic/elasticsearch-py/issues/1698 for ror information res_news = client.search(index = "news", [{'source': {'idi: 'engadget', 'name': 'Engadget'},
	result /var/folders/vi/2gby2gx803c03c01ixssjc0000gn/P/ipykernel_3052/1096609150.py:12: DeprecationNamning: If the 'body' parameter is deprecated for the 'search' ANI and will be removed in a future version. Instead use ANI parameters directly. See https://github.com/elastic/elasticsearch-py/issues/1698 for ince information res_news = client.search(index = "news",
	/vas/toidess/vi/2g8y2gage802c001jzx_asjc0000gm/T/lgydernes_2b0b2/10906091b0.py:12: DeprecationNernin ; The 'body' purchastor is destrocated for the 'socreth' Ael and will be removed in a future version. asked use ATT parameters directly. See https://glthub.com/elastic/elasticsearch.py/isaues/1808 for it of biforect in the control of the transfer in the control of
	The Decor parameter is demonstrated for the 'measure' AND and will be record to struct worder. The Decor parameter is demonstrated for the 'measure' AND and will be record to struct worder. The Decor parameter is demonstrated for the 'measure' AND and will be record to a struct worder. The Internation The John of the Parameters directly. Soon bitps://github.com/alastic/clianticamend-ps/isames/1630 for row_news_olsent.mostarity. "Internation" the parameters directly. Soon bitps://github.com/alastic/clianticamend-ps/isames/1630 for row_news_olsent. "Internation" "Apple acquests are exclusive deal with Tom Hanks' production company". "assortication: "Ample/Internation ministycer deal. The success of TV+ by locking in Tom Hanks' Name and assess and unscripted projects disable in set to start in a sequel to Noted Wall'. "Internation of the product of the parameter of the success of TV+ by locking in Tom Hanks' Name and assess and unscripted projects disable in the International Control of the Start of the International Control of t
	result Workfoldensewin/GapySepudOcoCcies: endcoCcOpyn/mispyternel 30015/100659155.py:12: approximatement or The Workfoldensewin/GapySepudOcoCcies: endcoCcOpyn/mispyternel 30015/100659155.py:12: approximatement or The Workfoldensewin in deprecated for the "secoch" ris and will be removed in a fatter eventor. Anterd are All To Generated Context. See things/faithLivens/elastin/mispytersewin/Secoch coverage into the Context. 1 incorvant: [1701] regardager, incorvi; "Disposer!], Incorvant: [1701] regardager, incorvi; "Disposer. Incorvant: [1701] regardager, incorvi; "Disposer. Incorvant: [1701] regardager, incorving the boat context on a sequel to World Main, Incorvant: [1701] regardager, incorving the boat context on a sequel to World Main, Incorvant: [1701] regardager, incorving the boat context on a sequel to World Main, Incorvant: [1702] regardager, incorving the boat context on a sequel to World Main, Incorpant: [1702] regardager, incorving the boat context on a sequel to World Main, Incorpant: [1702] regardager, incorving the boat context on a sequel to World Main, Incorpant: [1702] regardager, incorving the boat context on a sequel to World Main, Incorpant: [1702] regardager,
	would be set 20,000 to construct to decoursed for the "earth" for and while promoved in a future version, and and note AT promoved in a future version, and and note AT promoved in a future version, and and note AT promoved in a future version, and and note AT promoved in a future version, and and note AT promoved in a future version, and and note and and and are also promoved in a future version. The promoved in a future version of the promoved in a future version of the analysis of the an
	Treatment (1997) September 1 despendent for the 'scarce' AFT on File De communication of the 'scarce' AFT of the 'scarce'
	Available of the control of the cont
	reactions of 12-20-20-20-20-20-20-20-20-20-20-20-20-20
	The Contract of Christopher School in a metally approximate to the Color Parameter of the C
	months of the property of property of the service of the property of the prope
	The followed of all physiological and a company of the Manufagning and 1900 of 1900 and a company of the Manufagning and a company o
	The property of the control of the c
	mention of an electric problement of the property of the community of problement of the problement of the community of the co
[98]:	The control of the proposed pr
[98]:	The control of the co
[98]:	The control of the co
[98]:	The control of the co
[98]:	
[98]:	

In [130]:	<pre>import matplotlib.pyplot as plt xAxis = [i["date"] for i in result] yAxis = [float(i["close"]) for i in result] yAxis1 = [float(i["volume"]) for i in result] # Create figure and axis #1 fig, ax1 = plt.subplots() # plot line chart on axis #1 ax1.plot(xAxis,yAxis) ax1.set_ylabel('close') ax1.set_ylabel('close') ax1.set_ylim(min(yAxis)-2, max(yAxis)+2) ax1.legend(['Close Price(\$)'], loc="upper left") ax1.set_xlabel('Date') # set up the 2nd axis ax2 = ax1.twinx() # plot bar chart on axis #2 ax2.bar(xAxis,yAxis1, width=0.5, alpha=0.5, color='orange')</pre>
Out[130]:	# plot bar chart on axis #2 ax2.bar(xAxis,yAxis1, width=0.5, alpha=0.5, color='orange') ax2.grid(False) # turn off grid #2 ax2.set_ylabel('Volume') ax2.set_ylim(0, max(yAxis1)+10000000) ax2.legend(['Volume'], loc="upper right") <matplotlib.legend.legend 0x7fab21e18130="" at=""> le8 0.6 # Oose Price(\$) 0.8 0.6 # Oose Price(\$) 0.4</matplotlib.legend.legend>
In [123]: Out[123]:	174 - 0.2 173 - 0.2 2022-04-012022-03-312022-03-292022-03-282022-03-25 Date