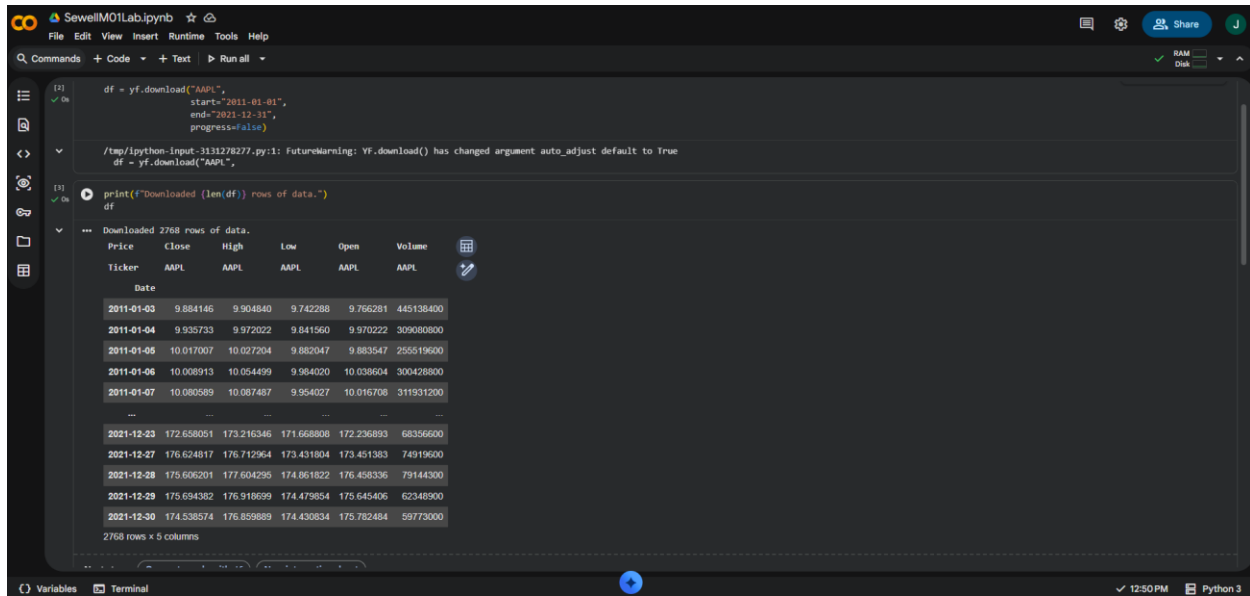


Yahoo Fiance



The screenshot shows a Jupyter Notebook interface with the following code and output:

```
df = yf.download("AAPL",
                  start="2011-01-01",
                  end="2021-12-31",
                  progress=False)

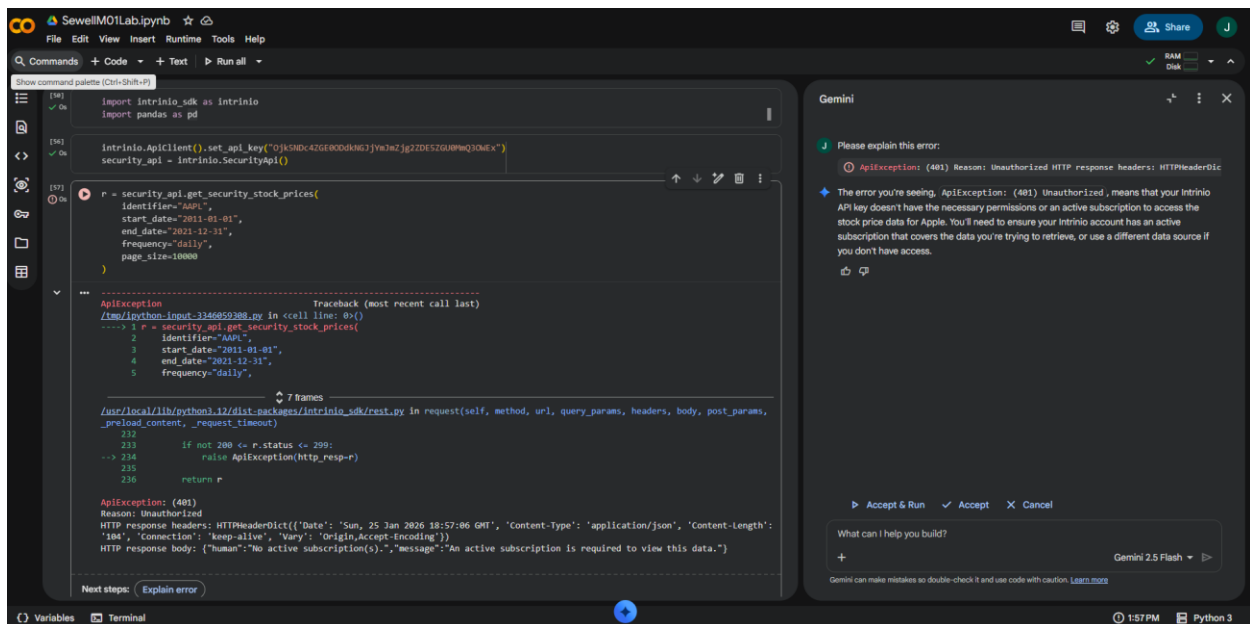
print(f"Downloaded {len(df)} rows of data.")
df
```

The output displays a table of stock data for Apple (AAPL) from 2011 to 2021. The table has 5 columns: Price, Close, High, Low, and Open. The data is grouped by date, showing daily price movements.

Date	Price	Close	High	Low	Open
2011-01-03	9.884146	9.904840	9.742288	9.766281	445138400
2011-01-04	9.935733	9.972022	9.841560	9.970222	309080800
2011-01-05	10.017007	10.027204	9.882047	9.883547	255519600
2011-01-06	10.009013	10.054499	9.984020	10.038604	300428800
2011-01-07	10.080589	10.087487	9.954027	10.016708	311931200
...
2021-12-23	172.658051	173.216346	171.668808	172.236893	68356600
2021-12-27	176.624817	176.712964	173.431804	173.451383	74919600
2021-12-28	175.605201	177.604295	174.861822	176.458336	79144300
2021-12-29	175.694382	176.918699	174.473654	175.645406	62348900
2021-12-30	174.538574	176.859889	174.430834	175.782484	59773000

2768 rows x 5 columns

Nasdaq Data Link. Gave me error message. After research this is No longer supported with free



The screenshot shows a Jupyter Notebook interface with the following code and output:

```
import intrinio_sdk as intrinio
import pandas as pd

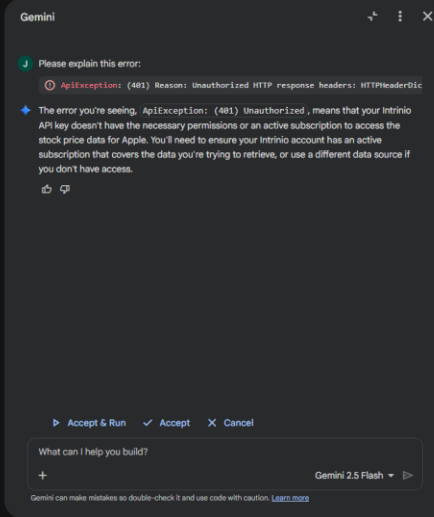
intrinio.ApiClient().set_api_key("0345NDx42G6800d8McJYv3e2Jg2ZDSZGURPwQ3OMeX")
security_api = intrinio.SecurityApi()

r = security_api.get_security_stock_prices(
    identifier="AAPL",
    start_date="2011-01-01",
    end_date="2021-12-31",
    frequency="daily",
    page_size=10000
)
```

The output shows a traceback for an `ApiException: (401) Reason: Unauthorized`. The error message indicates that the API key provided is not valid or that the account is not active.

```
ApiException: (401) Reason: Unauthorized
HTTP response headers: HTTPHeaderDict({'Date': 'Sun, 25 Jan 2026 18:57:06 GMT', 'Content-Type': 'application/json', 'Content-Length': '184', 'Connection': 'keep-alive', 'Vary': 'Origin,Accept-Encoding'})
HTTP response body: {"human": "No active subscription(s).", "message": "An active subscription is required to view this data."}
```

Next steps: [Explain error](#)



Gemini

Please explain this error:

ApiException: (401) Reason: Unauthorized HTTP response headers: HTTPHeaderDic

The error you're seeing, `ApiException: (401) Unauthorized`, means that your Intrinsic API key doesn't have the necessary permissions or an active subscription to access the stock price data for Apple. You'll need to ensure your Intrinsic account has an active subscription that covers the data you're trying to retrieve, or use a different data source if you don't have access.

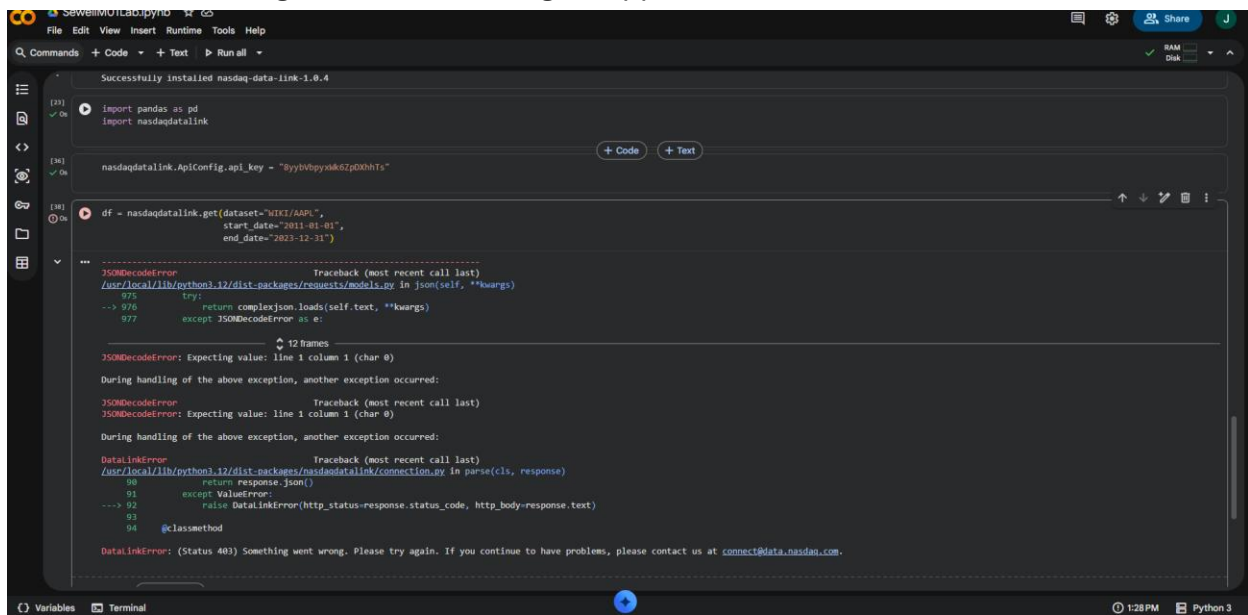
[Accept & Run](#) [Accept](#) [Cancel](#)

What can I help you build?

[+](#) [Gemini 2.5 Flash](#)

Gemini can make mistakes so double-check it and use code with caution. [Learn more](#)

Intrinio error message. This task No longer supported with free account



The screenshot shows a Jupyter Notebook interface with a dark theme. The code cell contains the following Python code:

```
Successfully installed nasdaq-data-link-1.0.4

[22] In [22]: import pandas as pd
import nasdaqdatalink

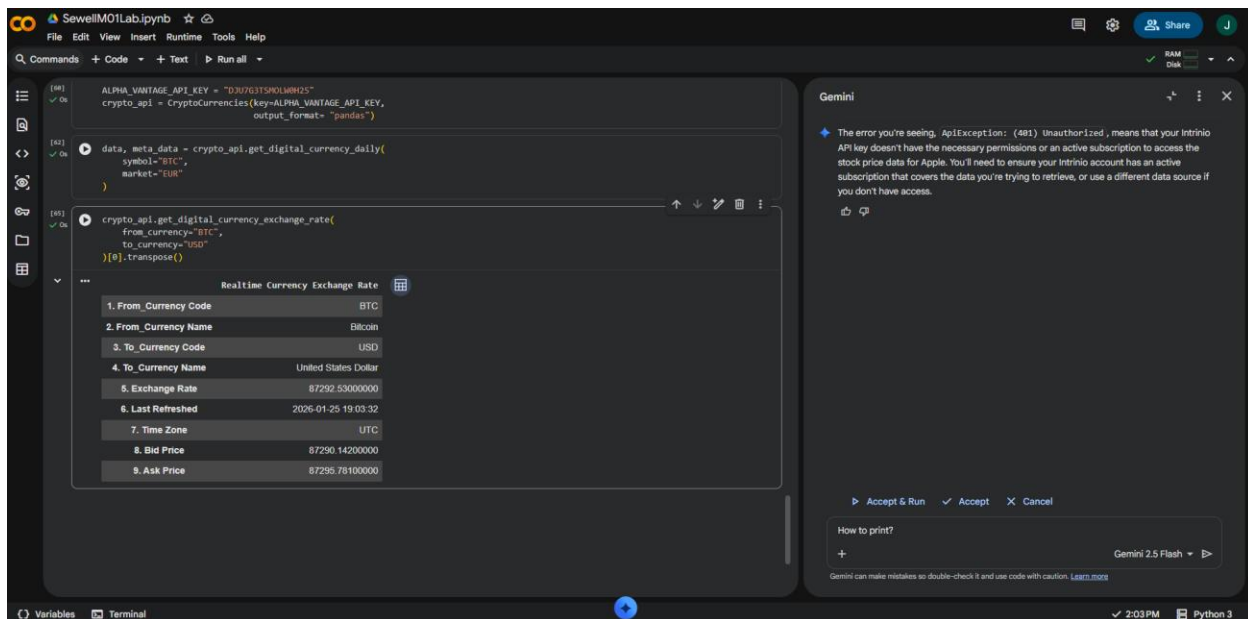
[36] In [36]: nasdaqdatalink.ApiConfig.api_key = "8yybvhpyxk67p00hhts"

[10] In [10]: df = nasdaqdatalink.get(dataset="NIKI/AAPL",
start_date="2011-01-01",
end_date="2023-12-31")

...
JSONDecodeError: Traceback (most recent call last)
/usr/local/lib/python3.12/dist-packages/requests/models.py in json(self, **kwargs)
975     try:
976         return complexjson.loads(self.text, **kwargs)
977     except JSONDecodeError as e:
...
12 frames
JSONDecodeError: Expecting value: line 1 column 1 (char 0)
During handling of the above exception, another exception occurred:
JSONDecodeError: Traceback (most recent call last)
JSONDecodeError: Expecting value: line 1 column 1 (char 0)
During handling of the above exception, another exception occurred:
DataLinkError: Traceback (most recent call last)
/usr/local/lib/python3.12/dist-packages/nasdaqdatalink/connection.py in parse(cls, response)
98     return response.json()
99     except ValueError:
100         raise DataLinkError(http_status=response.status_code, http_body=response.text)
93
94     @classmethod
DataLinkError: (Status 403) Something went wrong. Please try again. If you continue to have problems, please contact us at connect@data.nasdaq.com.
```

The error message indicates a 403 status code, which means the API key is not valid or the account is not authorized to access the data.

Alpha Vantage



The screenshot shows a Jupyter Notebook interface with a dark theme. The code cell contains the following Python code:

```
ALPHA_VANTAGE_API_KEY = "D3U763TS0LMWH25"
crypto_api = CryptoCurrencies(key=ALPHA_VANTAGE_API_KEY,
output_format="pandas")

[62] In [62]: data, meta_data = crypto_api.get_digital_currency_daily(
symbol="BTC",
market="EUR"
)

[10] In [10]: crypto_api.get_digital_currency_exchange_rate(
from_currency="BTC",
to_currency="USD"
)[0].transpose()

...
Realtime Currency Exchange Rate
1. From_Currency Code BTC
2. From_Currency Name Bitcoin
3. To_Currency Code USD
4. To_Currency Name United States Dollar
5. Exchange Rate 87292.53000000
6. Last Refreshed 2026-01-25 19:03:32
7. Time Zone UTC
8. Bid Price 87290.14200000
9. Ask Price 87295.78100000
```

The Gemini chat window is open on the right side of the notebook. It contains the following text:

The error you're seeing, `ApiException: (401) Unauthorized`, means that your Intrinio API key doesn't have the necessary permissions or an active subscription to access the stock price data for Apple. You'll need to ensure your Intrinio account has an active subscription that covers the data you're trying to retrieve, or use a different data source if you don't have access.

Accept & Run Accept Cancel

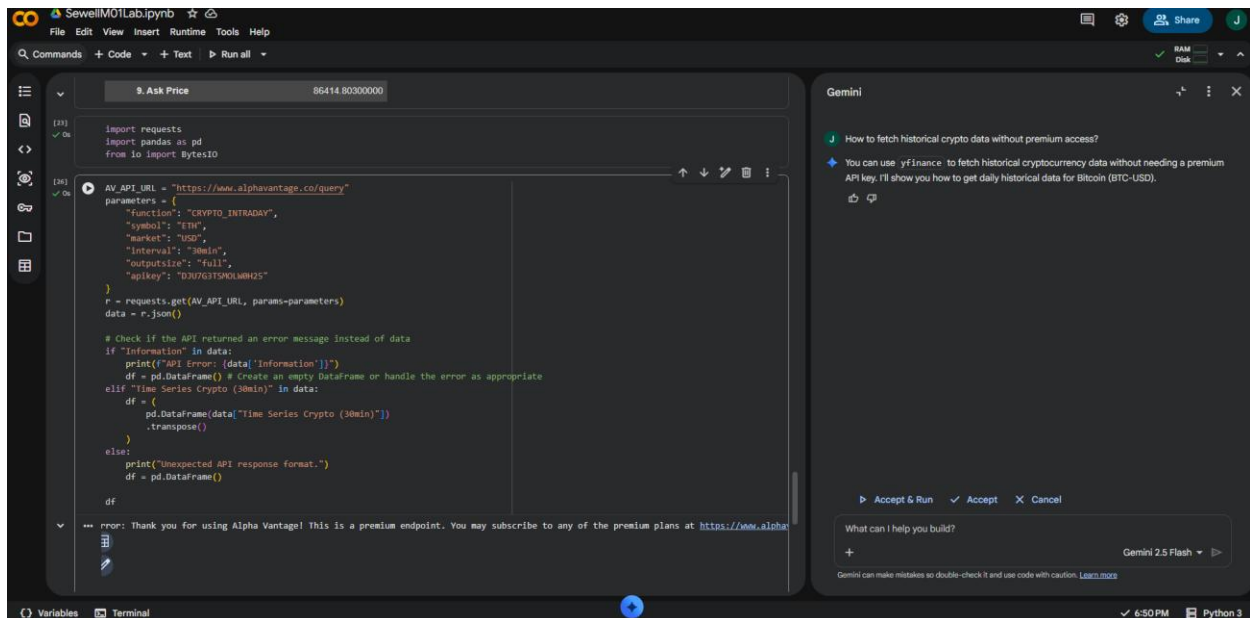
How to print?

+

Gemini 2.5 Flash

Gemini can make mistakes so double-check it and use code with caution. [Learn more](#)

Alpha Vantage Theres more error message here as well. This is now a premium feature to view intraday data



```
[23] In [23]: import requests
import pandas as pd
from io import BytesIO

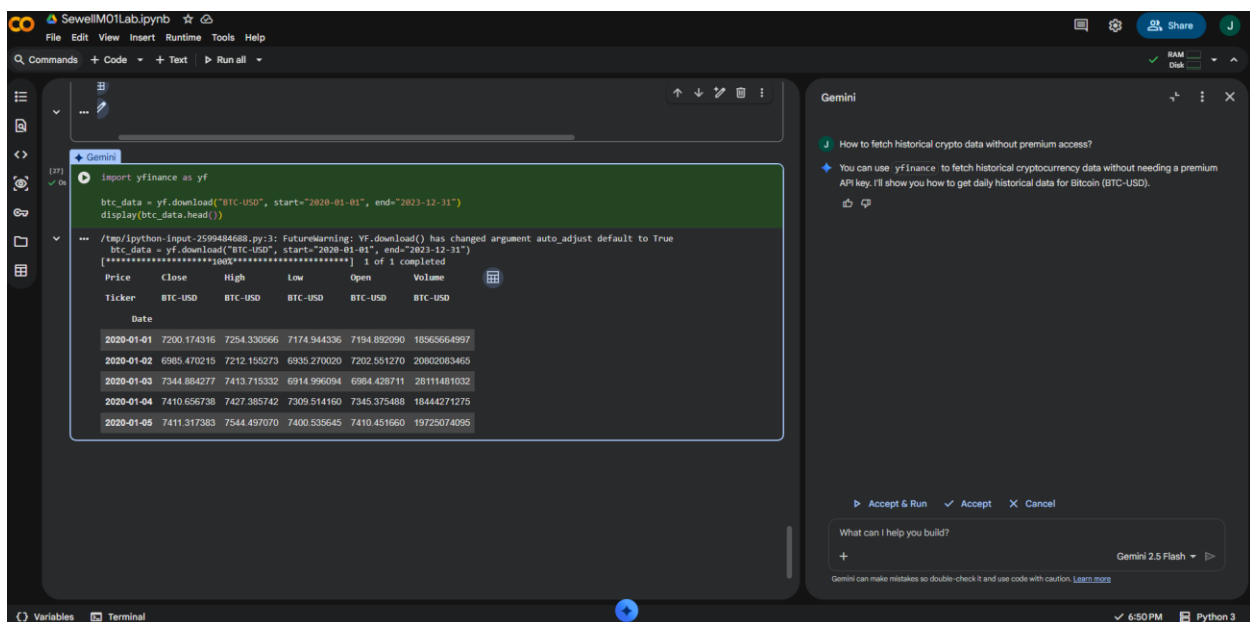
[24] In [24]: AV_API_URL = "https://www.alphavantage.co/query"
parameters = {
    "function": "CRYPTO_INTRADAY",
    "symbol": "ETH",
    "market": "USD",
    "interval": "30min",
    "outputsize": "full",
    "apikey": "DJU7G3T5NOLMWH25"
}
r = requests.get(AV_API_URL, params=parameters)
data = r.json()

# Check if the API returned an error message instead of data
if "Information" in data:
    print(f"API Error: {data['Information']}")
df = pd.DataFrame() # Create an empty DataFrame or handle the error as appropriate
elif "Time Series Crypto (30min)" in data:
    df = (
        pd.DataFrame(data["Time Series Crypto (30min)"])
        .transpose()
    )
else:
    print("Unexpected API response format.")
    df = pd.DataFrame()

df
```

Error: Thank you for using Alpha Vantage! This is a premium endpoint. You may subscribe to any of the premium plans at <https://www.alphavantage.com/subscribe/>

Able to use YFinance to view Bitcoin intraday Data



```
[27] In [27]: import yfinance as yf

btc_data = yf.download("BTC-USD", start="2020-01-01", end="2023-12-31")
display(btc_data.head())

FutureWarning: Yf.download() has changed argument auto_adjust default to True
btc_data = yf.download("BTC-USD", start="2020-01-01", end="2023-12-31")

[28] In [28]:
```

Price	Close	High	Low	Open	Volume
Ticker	BTC-USD	BTC-USD	BTC-USD	BTC-USD	BTC-USD
Date					
2020-01-01	7200.174316	7254.330566	7174.944336	7194.852090	18565664997
2020-01-02	6985.470215	7212.155273	6935.270020	7202.551270	20802083465
2020-01-03	7344.884277	7413.715332	6914.996094	6984.428711	20111481032
2020-01-04	7410.656738	7427.385742	7309.514160	7345.375488	18444271275
2020-01-05	7411.317383	7544.497070	7400.535645	7410.451660	19725074095

Earning Information using Alpha Vantage

SewellM01Lab.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

```
2020-01-02 6985.470215 7212.155273 6935.270020 7202.551270 20802083465
2020-01-03 7344.884277 7413.715332 6914.996094 6964.428711 20111481032
2020-01-04 7410.656738 7427.385742 7309.514160 7345.375488 18444271275
2020-01-05 7411.317383 7544.497070 7400.535645 7410.451660 19725074095
```

```
[28]: AV_API_URL = "https://www.alphavantage.co/query"
      parameters = {
        "function": "EARNINGS_CALENDAR",
        "horizon": "3month",
        "apikey": "03J76315QOLW8QJ5"
      }
      r = requests.get(AV_API_URL, params=parameters)
      pd.read_csv(BytesIO(r.content))
```

	symbol	name	reportDate	fiscalDateEnding	estimate	currency	timeOfTheDay
0	BKR	BAKER HUGHES CO - CLASS A	2026-01-25	2025-12-31	0.6700	USD	post-market
1	AGNC	AGNC INVESTMENT CORP	2026-01-26	2025-12-31	0.3700	USD	post-market
2	AGYS	AGILYSYS INC	2026-01-26	2025-12-31	0.3200	USD	post-market
3	AKO-A	EMBOTELLADORA ANDINA S.A.	2026-01-26	2025-12-31	0.5627	USD	NaN
4	ALPB	ALPB	2026-01-26	2025-12-31	NaN	USD	NaN
...
9675	SRRRF	SRRRF	2026-04-24	2026-03-31	NaN	USD	NaN
9676	TBOXF	TBOXF	2026-04-24	2026-03-31	0.0900	USD	NaN
9677	WAMFF	WAMFF	2026-04-24	2026-03-31	-0.0100	USD	NaN
9678	WDFCF	WDFCF	2026-04-24	2026-03-31	NaN	USD	NaN
9679	YGTFF	YGTFF	2026-04-24	2026-03-31	NaN	USD	NaN

9680 rows x 7 columns

Variables Terminal

6:53 PM Python 3

Gemini

How to fetch historical crypto data without premium access?

You can use `yfinance` to fetch historical cryptocurrency data without needing a premium API key. I'll show you how to get daily historical data for Bitcoin (BTC-USD).

Accept & Run Accept Cancel

What can I help you build?

+ Gemini 2.5 Flash

Gemini can make mistakes so double-check it and use code with caution. [Learn more](#)

Bitcoin OHLC prices CoinGecko

SewellM01Lab.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

```
[33]: from pycoingecko import CoinGeckoAPI
      from datetime import datetime
      import pandas as pd

[34]: cg = CoinGeckoAPI()

[35]: ohlc = cg.get_coin_ohlc_by_id(
      id="bitcoin", vs_currency="usd", days="14"
      )
      ohlc_df = pd.DataFrame(ohlc)
      ohlc_df.columns = ["date", "open", "high", "low", "close"]
      ohlc_df["date"] = pd.to_datetime(ohlc_df["date"], unit="ms")
      ohlc_df
```

	date	open	high	low	close
0	2026-01-12 00:00:00	90569.0	90819.0	90245.0	90819.0
1	2026-01-12 04:00:00	90868.0	92356.0	90694.0	91818.0
2	2026-01-12 08:00:00	91684.0	92128.0	91283.0	91283.0
3	2026-01-12 12:00:00	91287.0	91425.0	90311.0	90562.0
4	2026-01-12 16:00:00	90521.0	91446.0	90129.0	91446.0
...
78	2026-01-25 04:00:00	89097.0	89168.0	88740.0	88746.0
80	2026-01-25 08:00:00	88825.0	88940.0	88545.0	88545.0
81	2026-01-25 12:00:00	88543.0	88601.0	88120.0	88413.0
82	2026-01-25 16:00:00	88393.0	88801.0	88314.0	88314.0
83	2026-01-25 20:00:00	88317.0	88317.0	86386.0	86395.0

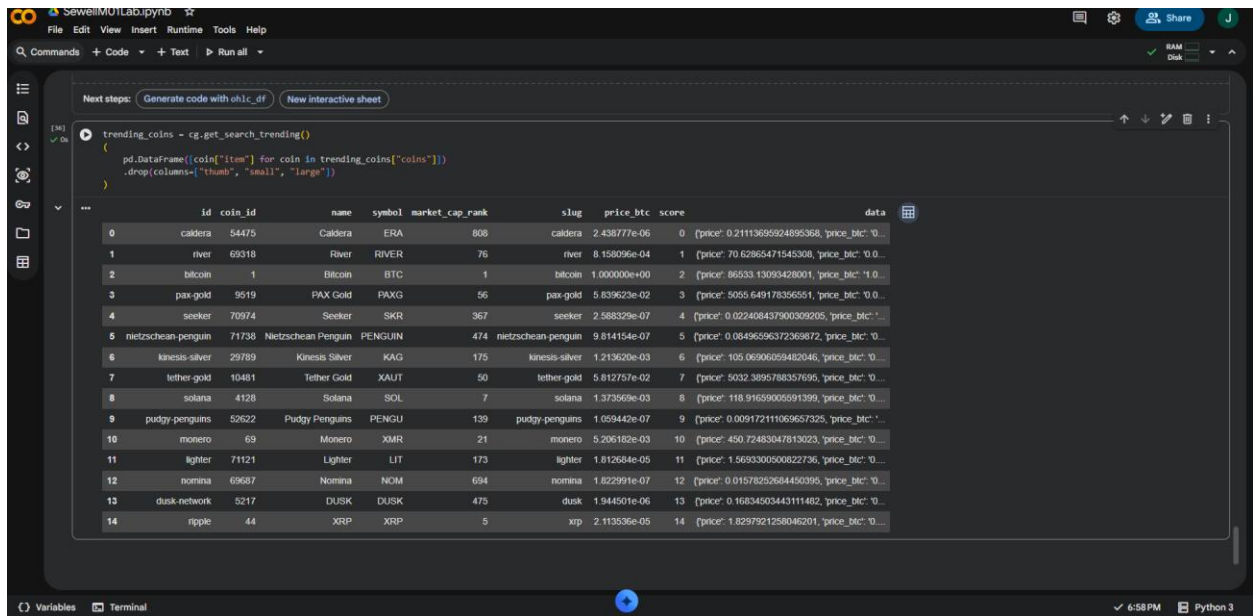
84 rows x 5 columns

Next steps: [Generate code with ohlc_df](#) [New interactive sheet](#)

Variables Terminal

6:56 PM Python 3

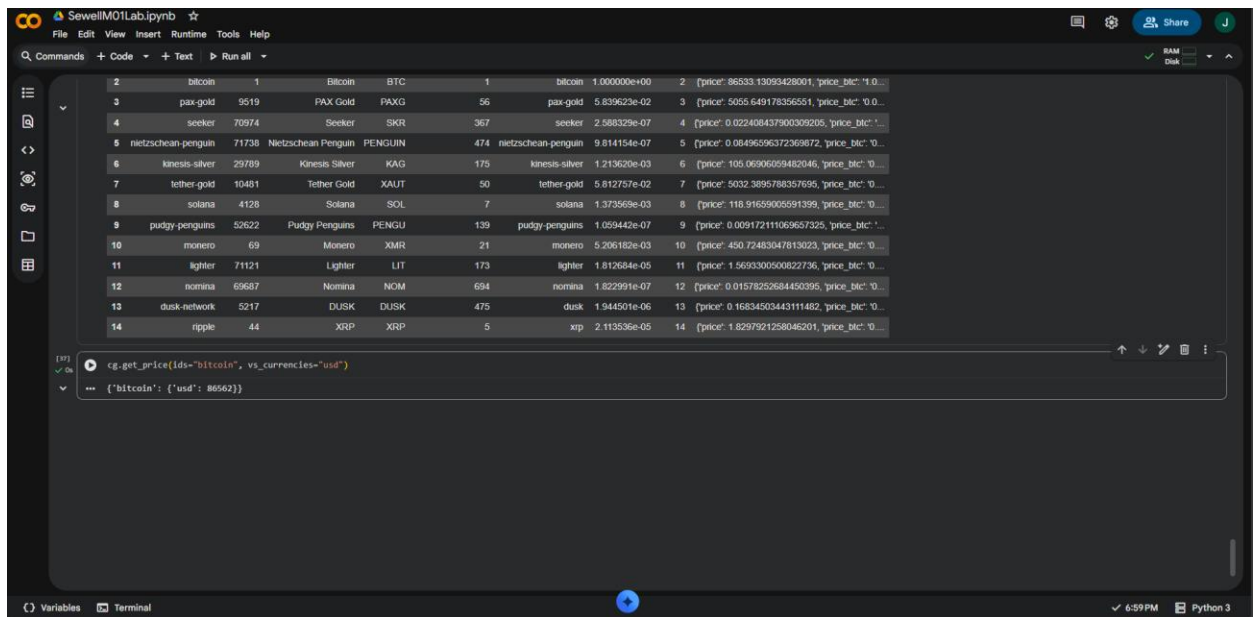
Top 7 trending coins Coin Gecko



```
trending_coins = cg.get_search_trending()
pd.DataFrame([coin["item"] for coin in trending_coins["coins"]])
.drop(columns=["thumb", "small", "large"])
```

	id	coin_id	name	symbol	market_cap_rank	slug	price_btc	score	data
0	caldera	54475	Caldera	ERA	808	caldera	2.43877e-06	0	{'price': 0.2111365924895368, 'price_btc': '0.0...
1	river	69318	River	RIVER	76	river	8.15809e-04	1	{'price': 70.62865471545308, 'price_btc': '0.0...
2	bitcoin	1	Bitcoin	BTC	1	bitcoin	1.000000e+00	2	{'price': 86533.13093428001, 'price_btc': '1.0...
3	pax-gold	9519	PAX Gold	PAXG	56	pax-gold	5.83962e-02	3	{'price': 5055.649178356551, 'price_btc': '0.0...
4	seeker	70974	Seeker	SKR	367	seeker	2.58832e-07	4	{'price': 0.022408437900309205, 'price_btc': '...
5	nietschean-penguin	71738	Nietschean Penguin	PENGUIN	474	nietschean-penguin	9.81415e-07	5	{'price': 0.08496596372369872, 'price_btc': '0.0...
6	kinesis-silver	29789	Kinesis Silver	KAG	175	kinesis-silver	1.21362e-03	6	{'price': 105.06906059482046, 'price_btc': '0.0...
7	tether-gold	10481	Tether Gold	XAUT	50	tether-gold	5.81275e-02	7	{'price': 5032.3895788357695, 'price_btc': '0.0...
8	solana	4128	Solana	SOL	7	solana	1.37356e-03	8	{'price': 118.91659005591399, 'price_btc': '0.0...
9	pudgy-penguins	52622	Pudgy Penguins	PENGU	139	pudgy-penguins	1.05944e-07	9	{'price': 0.009172111069657325, 'price_btc': '...
10	monero	69	Monero	XMR	21	monero	5.20618e-03	10	{'price': 450.72483047813023, 'price_btc': '0.0...
11	lighter	71121	Lighter	LIT	173	lighter	1.81268e-05	11	{'price': 1.5693300500822736, 'price_btc': '0.0...
12	nomina	69687	Nomina	NOM	694	nomina	1.82299e-07	12	{'price': 0.01578252684450395, 'price_btc': '0.0...
13	dusk-network	5217	DUSK	DUSK	475	dusk	1.94450e-06	13	{'price': 0.16834503443111482, 'price_btc': '0.0...
14	ripple	44	XRP	XRP	5	xrp	2.11353e-05	14	{'price': 1.8297921258046201, 'price_btc': '0.0...

Bitcoin current price coingecko



```
cg.get_price(ids="bitcoin", vs_currencies="usd")
```

```
{'bitcoin': {'usd': 86562}}
```