

Yahoo Finance

A screenshot of a Jupyter Notebook interface titled "Culley-M01Lab". The code cell contains Python code to download historical price data for Apple (AAPL) from 2011 to 2023. The output shows a DataFrame with columns: Date, Open, High, Low, Close, Volume, Dividends, Stock Splits. The data spans from January 3, 2011, to December 29, 2023. The final message indicates 3270 rows of data were downloaded.

```
aapl_data = yf.Ticker("AAPL")
aapl_data.history(
    start="2011-01-01",
    end="2023-12-31")
```

Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
2011-01-03 00:00:00-05:00	9.766283	9.904842	9.742290	9.884148	445138400	0.0	0.0
2011-01-04 00:00:00-05:00	9.970223	9.972023	9.841559	9.935734	309080800	0.0	0.0
2011-01-05 00:00:00-05:00	9.883547	10.027204	9.882047	10.017007	255519600	0.0	0.0
2011-01-06 00:00:00-05:00	10.038601	10.054496	9.984017	10.008910	300428800	0.0	0.0
2011-01-07 00:00:00-05:00	10.016709	10.087488	9.954028	10.080590	311931200	0.0	0.0
...
2023-12-22 00:00:00-05:00	193.353993	193.581852	191.164677	191.788788	37149600	0.0	0.0
2023-12-26 00:00:00-05:00	191.798670	192.076049	191.025969	191.243912	28919300	0.0	0.0
2023-12-27 00:00:00-05:00	190.689143	191.689688	189.302232	191.342957	48087700	0.0	0.0
2023-12-28 00:00:00-05:00	192.323695	192.838834	191.362768	191.768936	34049900	0.0	0.0
2023-12-29 00:00:00-05:00	192.085953	192.581275	189.936256	190.728775	42672100	0.0	0.0

3270 rows x 7 columns

Nasdaq Data Link

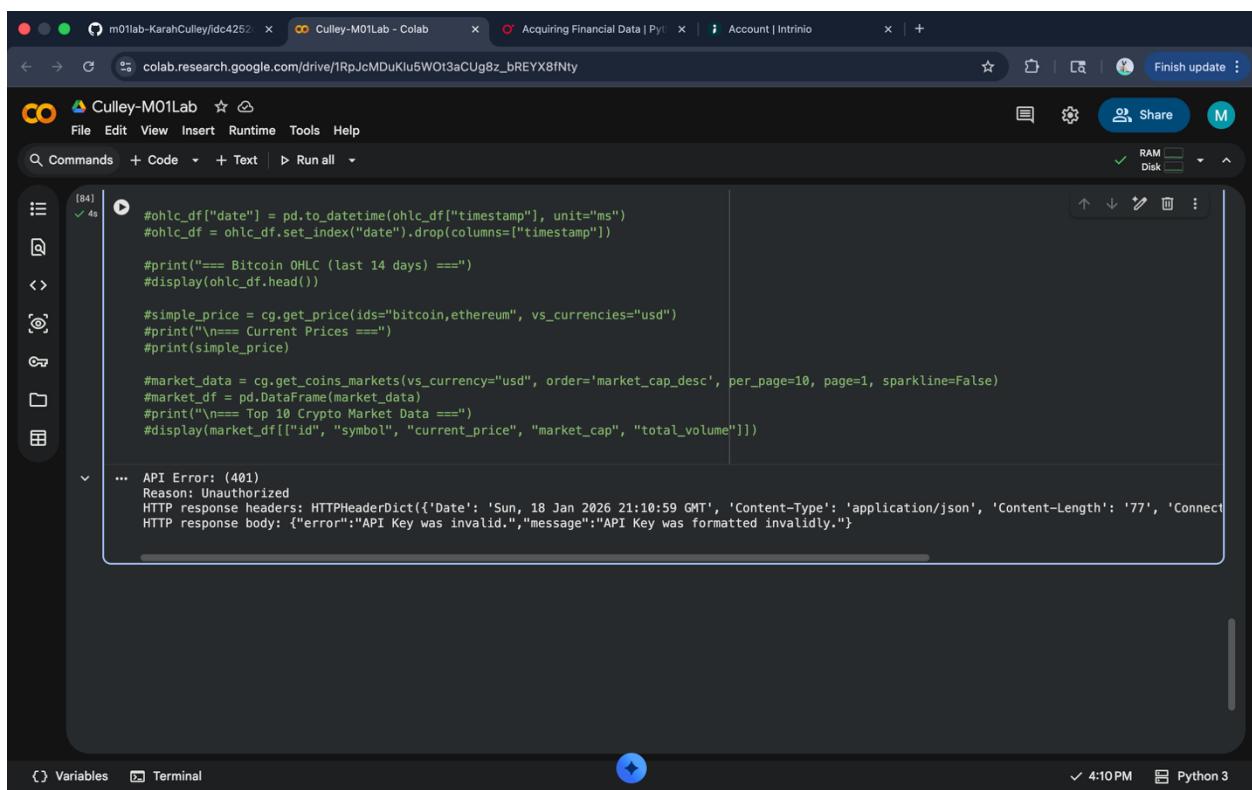
A screenshot of a Jupyter Notebook interface titled "Culley-M01Lab". The code cell contains Python code to display market data for AAPL. The output shows a DataFrame with columns: Price, Close, High, Low, Open, Volume. The data spans from January 3, 2011, to December 29, 2023. The final message indicates 1 of 1 completedDownloaded 2768 rows of AAPL data.

```
#print('==== top 10 Crypto Market Data ===')
#display(market_df[["id", "symbol", "current_price", "market_cap", "total_volume"]])
```

Price	Close	High	Low	Open	Volume
Ticker	AAPL	AAPL	AAPL	AAPL	AAPL
Date					
2011-01-03	9.884148	9.904842	9.742290	9.766283	445138400
2011-01-04	9.935732	9.972021	9.841559	9.970221	309080800
2011-01-05	10.017010	10.027207	9.882050	9.883550	255519600
2011-01-06	10.008908	10.054494	9.984016	10.038599	300428800
2011-01-07	10.080590	10.087488	9.954028	10.016709	311931200

Intrino

I tried various methods to get the Intrino part of the assignment to work only to receive error 401. I also tried using another API key to get the same error this is shown in the first screenshot. The second screenshot is pulling the data from Yahoo finance for Apples stock prices from 2011-2021 which the textbook was trying to do through Intrino.



The screenshot shows a Google Colab notebook titled "Culley-M01Lab". The code cell contains Python code for interacting with the cg API to get price data for Bitcoin and Ethereum, and to get market data for top 10 crypto markets. The output of the cell shows an API error 401:

```
#ohlc_df["date"] = pd.to_datetime(ohlc_df["timestamp"], unit="ms")
#ohlc_df = ohlc_df.set_index("date").drop(columns=["timestamp"])

#print("==== Bitcoin OHLC (last 14 days) ====")
#print(ohlc_df.head())

simple_price = cg.get_price(ids="bitcoin,ethereum", vs_currencies="usd")
#print("\n==== Current Prices ====")
#print(simple_price)

market_data = cg.get_coins_markets(vs_currency="usd", order='market_cap_desc', per_page=10, page=1, sparkline=False)
market_df = pd.DataFrame(market_data)
#print("\n==== Top 10 Crypto Market Data ====")
#print(market_df[["id", "symbol", "current_price", "market_cap", "total_volume"]])

...
... APT Error: (401)
Reason: Unauthorized
HTTP response headers: HTTPHeaderDict({'Date': 'Sun, 18 Jan 2026 21:10:59 GMT', 'Content-Type': 'application/json', 'Content-Length': '77', 'Connect': 'keep-alive'})
HTTP response body: {"error":"API Key was invalid.", "message":"API Key was formatted invalidly."}
```

```

#printf("\n==== Current Prices ===")
#print(simple_price)

#market_data = cg.get_coins_markets(vs_currency="usd", order='market_cap_desc', per_page=10, page=1, sparkline=False)
#market_df = pd.DataFrame(market_data)
#print("\n==== Top 10 Crypto Market Data ===")
#display(market_df[["id", "symbol", "current_price", "market_cap", "total_volume"]])

... [*****100%*****] 1 of 1 completedSUCCESS - historical price data retrieved

```

Date	Adj Close	AAPL	High	Low
2011-01-03 00:00:00	9.884147644042969	11.770357131958008	11.795000076293945	11.601428985595703
2011-01-04 00:00:00	9.935731887817383	11.831786155700684	11.875	11.719642639160156
2011-01-05 00:00:00	10.017009735107422	11.928570747375488	11.940713882446289	11.76785659790039
2011-01-06 00:00:00	10.0089082717895	11.918929100036621	11.973214149475098	11.889286041259766
2011-01-07 00:00:00	10.08050924810791	12.00428581237793	12.012499809265137	11.853570938110352

Show 25 per page

Like what you see? Visit the [data table notebook](#) to learn more about interactive tables.

Alpha Vantage

I tried various methods to get the Alpha Vantage part of the assignment to work only to receive errors. I also tried it with various API keys from their site the photos show what my screen looked like. I apologize for this inconvenience.

```

print(f"Alpha Vantage API Error: {data['Error Message']}")
elif "Time Series Crypto (30min)" in data:
    df = (
        pd.DataFrame(data["Time Series Crypto (30min)"])
        .transpose()
    )
    print("Successfully retrieved Alpha Vantage crypto data.")
    print(df.head())
else:
    print("Unexpected Alpha Vantage API response format.")
    print(data)

... /tmp/ipython-input-3048813576.py:4: FutureWarning: YF.download() has changed argument auto_adjust default to True
      df = yf.download("AAPL",
Downloaded 2768 rows of data.

KeyError                                     Traceback (most recent call last)
/tmp/ipython-input-3048813576.py in <cell line: 0>()
      54 data = r.json()
      55 df = (
--> 56     pd.DataFrame(data["Time Series Crypto (30min)"])
      57     .transpose()
      58 )

KeyError: 'Time Series Crypto (30min)'

Next steps: Explain error

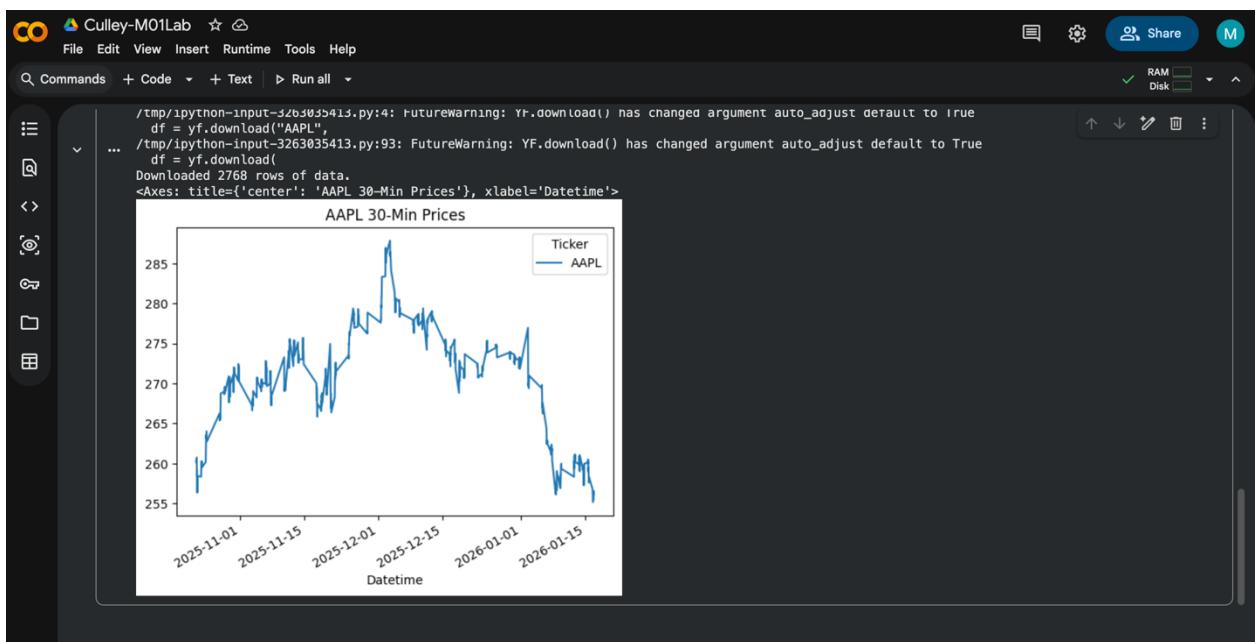
```

```
AV_API_URL = "https://www.alphavantage.co/query"
params = {
    "function": "TIME_SERIES_INTRADAY",
    "symbol": "AAPL",
    "interval": "30min",
    "outputsize": "compact",
    "apikey": ALPHA_VANTAGE_API_KEY
}

r = requests.get(AV_API_URL, params=params)
data = r.json()
data
```

... /tmp/ipython-input-123586991.py:4: FutureWarning: YF.download() has changed argument auto_adjust default to True
df = yf.download("AAPL",
Downloaded 2768 rows of data.
{'Information': 'Thank you for using Alpha Vantage! This is a premium endpoint. You may subscribe to any of the premium plans at https://www.alphavantage.co/premium/'}

This is some data from Yahoo for AAPL stock. It's not what Alpha Vantage would have retrieved but it shows the 30-minute interval price changes for the stock over the 60-day period.



Coin Gecko

The screenshot shows a Jupyter Notebook interface with the title "Culley-M01Lab". The code cell displays the following output:

```
==== Bitcoin OHLC (last 14 days) ====
date
2026-01-05 00:00:00 91104.0 91428.0 91104.0 91373.0
2026-01-05 04:00:00 91586.0 93169.0 91586.0 92875.0
2026-01-05 08:00:00 92858.0 92872.0 92150.0 92465.0
2026-01-05 12:00:00 92481.0 93073.0 92301.0 93073.0
2026-01-05 16:00:00 93083.0 93841.0 92475.0 93825.0

==== Current Prices ====
{'bitcoin': {'usd': 95284}, 'ethereum': {'usd': 3340.36}}

==== Top 10 Crypto Market Data ====
id symbol current_price market_cap total_volume
0 bitcoin btc 95288.000000 1904076203386 19338305685
1 ethereum eth 3340.450000 403395042340 11859000563
2 tether usdt 0.999636 186924385561 37654945383
3 binancecoin bnb 951.860000 129813529003 1152642869
4 ripple xrp 2.060000 124971638366 1170187603
5 solana sol 142.360000 80497478191 2578259802
6 usd-coin usdc 0.999701 75949846400 2778575125
7 tron trx 0.320372 30336964635 623159874
8 staked-ether steth 3340.320000 30043113752 10339609
```

The screenshot shows a Google Colab interface with the title "Culley-M01Lab". The code cell displays the same output as the Jupyter Notebook:

```
==== Bitcoin OHLC (last 14 days) ====
date
2026-01-05 00:00:00 91104.0 91428.0 91104.0 91373.0
2026-01-05 04:00:00 91586.0 93169.0 91586.0 92875.0
2026-01-05 08:00:00 92858.0 92872.0 92150.0 92465.0
2026-01-05 12:00:00 92481.0 93073.0 92301.0 93073.0
2026-01-05 16:00:00 93083.0 93841.0 92475.0 93825.0

==== Current Prices ====
{'bitcoin': {'usd': 95284}, 'ethereum': {'usd': 3340.36}}

==== Top 10 Crypto Market Data ====
id symbol current_price market_cap total_volume
0 bitcoin btc 95288.000000 1904076203386 19338305685
1 ethereum eth 3340.450000 403395042340 11859000563
2 tether usdt 0.999636 186924385561 37654945383
3 binancecoin bnb 951.860000 129813529003 1152642869
4 ripple xrp 2.060000 124971638366 1170187603
5 solana sol 142.360000 80497478191 2578259802
6 usd-coin usdc 0.999701 75949846400 2778575125
7 tron trx 0.320372 30336964635 623159874
8 staked-ether steth 3340.320000 30043113752 10339609
9 dogecoin doge 0.137382 23129892199 538396541
```

