dataset_creation

January 31, 2026

– Data Ingestion –

[1]:

```
#Go to repo
%cd /home/jupyter-toomeyck/HelpHerInvest
```

```
/home/jupyter-toomeyck/HelpHerInvest
```

[2]:

```
#Sync latest from GitHub before editing
!git pull --rebase origin main
```

```
remote: Enumerating objects: 263, done.
remote: Counting objects: 100% (263/263), done.
remote: Compressing objects: 100% (228/228), done.
remote: Total 260 (delta 137), reused 50 (delta 24), pack-reused 0 (from 0)
Receiving objects: 100% (260/260), 599.51 KiB | 12.23 MiB/s, done.
Resolving deltas: 100% (137/137), completed with 1 local object.
From https://github.com/tongyuguo/HelpHerInvest
 * branch            main       -> FETCH_HEAD
   6f2d429..542dd31  main       -> origin/main
CONFLICT (file location): Week 01 Identify the Problem Statement and
Dataset/Test.ipynb added in 786bbac (Submit) inside a directory that was renamed
in HEAD, suggesting it should perhaps be moved to Playground/Week 01 Identify
the Problem Statement and Dataset/Test.ipynb.
error: could not apply 786bbac… Submit
hint: Resolve all conflicts manually, mark them as resolved with
hint: "git add/rm <conflicted_files>", then run "git rebase --continue".
hint: You can instead skip this commit: run "git rebase --skip".
hint: To abort and get back to the state before "git rebase", run "git

rebase --abort".
Could not apply 786bbac… Submit
```

[4]:

```
## Imports
# libraries

import time
import requests
import pandas as pd
#%pip install yfinance --quiet
```

```
import yfinance as yf
from pathlib import Path
import numpy as np
import warnings
warnings.filterwarnings("ignore")
```

[ ]:

```
## PARAMETERS ##
## CHANGE OUTPUT PATH ##

repo_root = Path("/home/jupyter-toomeyck/HelpHerInvest")
output_path = repo_root / "Data" / "stock_symbols_new.csv.zip"
output_path.parent.mkdir(parents=True, exist_ok=True)
output_file = "stock_symbols_new.csv.zip"

SEC_URL = "https://www.sec.gov/files/company_tickers_exchange.json"
SEC_HEADERS = {"User-Agent": "YourAppName your_email@example.com"}  # required␣
 ↪by SEC


def get_universe_from_sec(limit):
    r = requests.get(SEC_URL, headers=SEC_HEADERS, timeout=30)
    r.raise_for_status()
    j = r.json()
    df = pd.DataFrame(j["data"], columns=j["fields"])
    df = df.rename(columns={"ticker": "symbol", "name": "company_name"})
    df["symbol"] = df["symbol"].str.upper()
    return df[["symbol", "company_name"]].drop_duplicates()

def yf_fetch_info(symbol: str) -> dict:
    # Normalize common Yahoo symbol formatting
    # BRK-B on SEC often needs BRK-B or BRK.B depending; yfinance likes BRK-B␣
 ↪*sometimes* but BRK.B often works.
    # We'll try a small fallback.
    candidates = [symbol, symbol.replace("-", ".")]
    for sym in candidates:
        try:
            t = yf.Ticker(sym)
            info = t.get_info()  # yfinance >= 0.2.0 style
            if info and isinstance(info, dict) and info.get("quoteType") in␣
 ↪("EQUITY", "ETF"):
                return info
        except Exception:
            pass

    return info

def build_base_table(limit, sleep_s=0.35):
    universe = get_universe_from_sec(limit=limit)
    print("Symbols pulled:",len(universe.index))
    rows = []
    count = 0
    for sym in universe["symbol"].tolist():
        rows.append(yf_fetch_info(sym))
        count += 1
        time.sleep(sleep_s)  # throttle to avoid Yahoo blocks
        if count % 200 == 0:
            print("{} rows completed".format(count))

    facts = pd.DataFrame(rows)

    df_cols = list(facts.columns)
    added = ["symbol", "company_name"]
    cols = df_cols + added

    base = (
        universe
        .merge(facts, on="symbol", how="left")
        [cols]
        .drop_duplicates(subset=["symbol"])
    )

    return base

df_base = build_base_table(limit=1000, sleep_s=0.35)
#universe = get_universe_from_sec(limit=2000)
#print(universe)
#df_base = pd.DataFrame()

print(df_base.head(10))
print(df_base.shape)
print(df_base.columns)

## CHANGE THE OUTPUT PATH ##

df_base.to_csv(output_path,index=False)
df = pd.read_csv(output_path)

grouped = df.groupby("sector")["sector"].count()
print(grouped)
```

```
Symbols pulled: 10338
```