Milestone 1: Data Pipeline

QM 2023 Capstone Project

Due: Friday, Week 5 (February 20, 2026) by 11:59 PM **Points:** 50 (25% of capstone grade) **Format:** Team submission via GitHub Classroom

Important: Work in your team's **shared private GitHub repository** (the same repo you'll use for all capstone milestones). Do not create separate repositories for different milestones.

Overview

Build the **foundational data infrastructure** for your capstone: integrate your primary dataset (REIT returns by default, or approved alternative) with supplementary data (economic indicators, policy measures, or market factors) into a clean, analysis-ready panel.

Real-world context: Professional analysts spend 60–80% of their time on data engineering. A robust pipeline ensures all subsequent analysis runs smoothly and reproducibly.

Success criterion: Another analyst (or your instructor) should be able to run your script from scratch and obtain identical results.

Directory Structure

M1 outputs go to `data/final/`, not `results/`. Use `config_paths.py` for consistent paths:

- **Input:** Primary data from `data/raw/`; supplementary data as needed
- **Output:** Analysis-ready panel to `data/final/[dataset]_analysis_panel.csv` and `data/final/[dataset]_metadata.json`

See `STRUCTURE.md` in the Capstone Project folder for the full folder guide.

Learning Objectives

1. **Fetch or load data** from appropriate sources (APIs, databases, CSV files)
2. **Clean messy datasets** (missing values, duplicates, outliers, date formatting)
3. **Merge datasets** on common keys (Date/Time, Entity ID) while preserving data integrity
4. **Reshape data** into tidy panel structure (Entity × Time)
5. **Document data decisions** (missing values, outliers, filters, transformations)
6. **Produce reproducible output** (relative paths, clear comments, metadata)

Deliverables

1. Python Script: `capstone_data_pipeline.py`

Requirements:

- Runs without errors from top to bottom
- Uses **relative paths only** (no hardcoded `C:\Users\...`)
- Includes clear section headers and comments explaining each step

- Outputs a tidy CSV file ready for analysis

Core Sections:

```
"""
QM 2023 Capstone Project: Milestone 1 Data Pipeline
Team: [Your Team Name]
Members: [List names]
Date: [Submission date]
Dataset: [REIT Master / Cryptocurrency / Housing / Other]

This script fetches, cleans, and merges [your dataset] into a tidy panel structure.
"""

# Section 1: Imports and setup
# Section 2: Load primary dataset
# Section 3: Clean data (missing values, outliers, duplicates)
# Section 4: Fetch/load supplementary data (economic indicators, policy measures)
# Section 5: Merge primary + supplementary data
# Section 6: Reshape to panel structure (Entity × Time)
# Section 7: Save tidy output and metadata
```

---

2. Tidy Output File: `[dataset_name]_analysis_panel.csv`

Generic Panel Structure:

[entity_id]	[time_var]	[outcome]	[char_1]	[char_2]	[driver_1]	[driver_2]	[control_1]	[control_2]
[ID_1]	[T_1]	[Y_1]	[X_1]	[X_2]	[Z_1]	[Z_2]	[C_1]	[C_2]

REIT: permno | ym | ret | mcap | sector | price | fedfunds | mortgage30us | cpiaucsl | unrate
Crypto: token_id | date | return_pct | mcap_usd | token_type | volume_usd | reg_severity | btc_return | cpi | vix
Housing: fips_code | year_quarter | rent_growth_yoy | median_price | region | income_median | mortgage_rate | gdp_growth | hpi | unrate

Requirements:

- **Long format** (one row per entity-time observation)
- **No missing keys** (entity_id and time_var must be non-null for all rows)
- **Date consistency** (time_var: `YYYY-MM` monthly, `YYYY-MM-DD` daily, `YYYY-QX` quarterly)
- **Merged correctly** (supplementary data aligned to time, no duplicate rows)

---

3. Data Quality Report: `M1_data_quality_report.md`

Required sections:

1. **Data Sources** — Primary: name, source, coverage (N entities, frequency, date range), initial row count, key variables. Supplementary: source, variables, date range.
2. **Data Cleaning Decisions** — For each: variable, % missing/count, decision (drop/impute/cap/winsorize), justification. Cover: missing values, outliers, size/volume filters, duplicates, data type corrections.
3. **Merge Strategy** — Join type (left/inner), keys, alignment. Before/after row counts; verification that counts make sense.

4. **Final Dataset Summary** — Entity variable, time variable, balanced/unbalanced, final dimensions. Sample statistics table (mean, std, min, max, missing %). Data quality flags.
5. **Reproducibility Checklist** — Script runs, relative paths, output location, no manual editing, metadata + AI Audit complete.
6. **Ethical Considerations** — What data are we losing? Who might we exclude? Example: "*By dropping mcap <$10M REITs, we exclude micro-caps. Acceptable for institutional focus; small REITs have different risk profiles.*" Transparency; test alternatives in M3 robustness.

Example entries (adapt to your dataset):

Decision	REIT	Crypto
Missing outcome	Drop (5.2%—delistings)	Drop (2.1%—new listings)
Outliers	Winsorize 99th/1st pct	Cap ±100%
Size filter	Drop mcap < $10M	Drop tokens <30 days history
Duplicates	Keep first	Average price, sum volume

Worked REIT example (Section 1–4):

- **Primary:** REIT Master Panel, instructor-provided. 532 REITs, monthly 2015-01–2023-12, 56,800 rows. Keys: permno, ym, ret, mcap, sector, price.
- **Supplementary:** FRED (FEDFUNDS, MORTGAGE30US, CPIAUCSL, UNRATE) via pandas-datareader, 108 months.
- **Cleaning:** ret 5.2% missing → drop (delistings); sector 3.1% → drop; returns >200% or <-100% → winsorize 99/1 pct; mcap <$10M → drop; duplicate permno-ym → keep first; ym "2020m1" → "2020-01".
- **Merge:** Left join on month; 48,258 rows before and after; 0 NaNs in supplementary.
- **Final:** 532 REITs × 108 months = 48,258 obs (unbalanced).
- **Sample stats table:** Variable | Mean | Std Dev | Min | Max | Missing (%). Fill for outcome, chars, drivers, controls.

Sign-off: All team member names.

---

4. AI Audit Appendix: `AI_AUDIT_APPENDIX.md`

Required for all milestones. Document AI use with "Disclose, Verify, Critique":

```
## AI Tools Used
- [ ] ChatGPT / Copilot / Claude / Other

## Per Task
- **Task:** [Description]
- **Prompt:** "[Exact prompt]"
- **AI Output:** [Code/text provided]
- **Verification:** [How you tested]
- **Critique:** [Right/wrong, your corrections]

## Summary
Total AI use, primary use cases, verification method.
Responsibility: All code is tested and our responsibility.
```

No AI Audit Appendix = 0/50 (enforced strictly). See M4 memo template for detailed examples.

Technical Requirements

1. Data Sources

REIT Teams (Default):

- **Primary:** `data/reit_master_raw.csv` — permno, ym, ret, mcap, sector, price. Known issues: ~5% missing returns, ~2% duplicate permno-month, outliers, inconsistent ym formatting.
- **Factors:** `data/reit_factors.csv` — ym, SIZE, VALUE, MOM, QLTY, LOWVOL, REV, ewtret, vwtret (clean).
- **Supplementary:** FRED via `pandas-datareader` — FEDFUNDS, MORTGAGE30US, CPIAUCSL, UNRATE (match REIT date range).

Alternative Dataset Teams: Principles: Entity × Time structure; supplementary drivers/controls; merge on time frequency; document all cleaning decisions.

---

2. Data Cleaning Pipeline (Generic Pattern)

Step 1 — Load & inspect:

```
primary_df = pd.read_csv(RAW_DATA_DIR / '[your_dataset].csv')
primary_df.columns = primary_df.columns.str.lower().str.replace(' ', '_')
print(primary_df.shape, primary_df.head(), primary_df.info())
```

Step 2 — Parse time:

```
# Monthly: primary_df['ym'] = pd.to_datetime(primary_df['ym'].str.replace('m', '-'),
errors='coerce')
# Daily:   primary_df['date'] = pd.to_datetime(primary_df['timestamp'], unit='s')
```

Step 3 — Missing values: Document before (`df.isnull().sum()`). Typically drop outcome; justify imputation.

```
print(f"Missing outcome: {primary_df['outcome'].isnull().sum()}
({100*primary_df['outcome'].isnull().mean():.1f}%)")
primary_clean = primary_df.dropna(subset=['outcome'])
```

Step 4 — Duplicates: `drop_duplicates(subset=['entity_id','time_var'], keep='first')` (or average/sum per domain).

Step 5 — Outliers: Winsorize, cap, or drop. Justify by domain.

```
upper, lower = primary_clean['outcome'].quantile(0.99),
primary_clean['outcome'].quantile(0.01)
primary_clean['outcome'] = primary_clean['outcome'].clip(lower=lower, upper=upper)
# Crypto/high-vol: primary_clean['outcome'] = primary_clean['outcome'].clip(-100, 100)
```

Step 6 — Size/volume filters: Drop entities below threshold (e.g., mcap < $10M). Document.

---

3. Fetch Supplementary Data

FRED (REIT, Housing, Macro):

```
import pandas_datareader as pdr
fedfunds = pdr.DataReader('FEDFUNDS', 'fred', start, end)
mortgage = pdr.DataReader('MORTGAGE30US', 'fred', start, end)
# ... concat, reset_index, rename columns. Align time to match primary (e.g.,
.dt.to_period('M')).
```

Events/Custom: Load CSV, ensure datetime, fill missing dates (e.g., severity=0 for no event).

---

4. Merge Strategy

Align supplementary time to primary. Use **left join**. Fill missing: forward fill, 0 for "no event," or drop.

```
# Align time: econ_df['ym'] = pd.to_datetime(econ_df['DATE']).dt.to_period('M')
merged = primary_clean.merge(econ_df, on='ym', how='left')

# Critical check
assert merged.shape[0] == primary_clean.shape[0], "Row count mismatch!"
# If merged >> primary: many-to-many. supp_df.drop_duplicates(subset=['time_var']) before
merge
```

Panel verification:

```
print(merged['entity_id'].nunique(), merged['time_var'].nunique(), merged.shape[0])
obs_per_entity = merged.groupby('entity_id')['time_var'].count()
print(obs_per_entity.min(), obs_per_entity.max())  # unbalanced if min ≠ max
```

---

5. Save Output

```
merged.to_csv(FINAL_DATA_DIR / '[dataset]_analysis_panel.csv', index=False)
metadata = {
    'dataset': '[name]', 'n_entities': int(merged['entity_id'].nunique()),
    'n_time_periods': int(merged['time_var'].nunique()), 'n_obs': int(merged.shape[0]),
    'time_range': f"{merged['time_var'].min()} to {merged['time_var'].max()}",
    'variables': list(merged.columns),
    'cleaning_decisions': {'missing_outcome': 'Dropped', 'outliers': 'Winsorized',
'size_filter': '...'}
}
with open(FINAL_DATA_DIR / '[dataset]_metadata.json', 'w') as f:
    json.dump(metadata, f, indent=2, default=str)
```

Grading Rubric (50 points)

Component	Points	Criteria
Reproducibility	15	Script runs; relative paths; clear section headers and comments
Data Cleaning	12	Missing, duplicates, outliers handled; before/after counts documented
Merge Integrity	8	No row loss/duplication; supplementary data aligned correctly
Panel Structure	8	Correct entity-time format; verified dimensions
Documentation	7	Data quality report complete; metadata JSON; cleaning decisions justified

Zero-Credit Conditions:

- Missing `AI_AUDIT_APPENDIX.md` = **0/50**
- Script won't run (syntax/path errors) = **0/50**
- Hardcoded absolute paths = **-10 points**

Common Pitfalls

Pitfall	Problem	Solution
Hardcoded paths	`C:/Users/...` won't run on other machines	Use `config_paths` and relative paths
Merge duplicates	merged rows 10× primary	One row per time in supplementary; deduplicate before merge
Time type mismatch	string vs datetime; NaT after merge	`pd.to_datetime()` on both datasets
No cleaning docs	"Cleaned data" with no details	Document every step: counts + economic justification
Missing AI Audit	No appendix	**0/50** — log AI use as you work
Wrong thresholds	REIT winsorize applied to crypto	Use domain-appropriate thresholds for your data

Testing Checklist (Run Before Submission)

- ☐ Script runs from scratch (delete `data/final/`, rerun)
- ☐ No absolute paths (search for `C:\`, `/Users/`)
- ☐ No manual Excel editing — script alone produces output
- ☐ Row counts: merged ≤ primary (no accidental duplication)
- ☐ Spot-check supplementary values (e.g., FEDFUNDS Jan 2020 ~1.55%)
- ☐ All deliverables: `capstone_data_pipeline.py`, `*_analysis_panel.csv`, `*_metadata.json`, `M1_data_quality_report.md`, `AI_AUDIT_APPENDIX.md`
- ☐ Team names in all files
- ☐ Alternative dataset approved (if applicable) by Week 4

Submission Instructions

1. Accept M1 assignment via GitHub Classroom
2. Clone repo, add files, commit, push:

```
git add capstone_data_pipeline.py data/final/*.csv data/final/*.json
M1_data_quality_report.md AI_AUDIT_APPENDIX.md
git commit -m "M1: Data Pipeline - [Dataset Name]"
git push origin main
```

3. Verify files visible on GitHub; GitHub Actions will run checks.

Deadline: Friday, Week 5 (Feb 20, 2026) 11:59 PM | **Late:** 10% per day, up to 3 days.

Resources and Support

- **Office Hours:** Dr. Seagraves, Mon & Wed 3–5 PM (Helm 122-D)
- **Starter script:** `starter/capstone_data_pipeline.py`
- **pandas-datareader:** Official docs

Debugging: Merge not working? Print both DataFrames; ensure keys match (type + values). API failing? FRED can rate-limit; try `time.sleep(1)` between requests.

Next Steps After M1

M2 (Week 9) — EDA and visualization. **Don't wait until Week 8 to start.** Common M2 discovery → M1 fix:

- M2 shows "too many missing in lag 2" → Add 2-period lag in M1
- M2 sector analysis fails → Verify sector cleaning in M1
- M2 correlations look strange → Check merge/date alignment in M1
