Milestone 1: Data Pipeline

QM 2023 Capstone Project

Due: Wednesday, February 25, 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.

Repository Structure

Your capstone repository should follow this modular structure:

```
QM-2023-Capstone-Repo/
├── code/
│   ├── config_paths.py              # Path management (provided)
│   ├── fetch_[dataset1]_data.py     # Fetch + clean primary dataset
│   ├── fetch_[dataset2]_data.py     # Fetch + clean supplementary dataset
│   ├── fetch_[dataset3]_data.py     # (Optional) Additional dataset
│   └── merge_final_panel.py         # Merge processed → final
├── data/
│   ├── raw/                         # Original downloaded data
│   │   ├── *_raw.csv
│   │   └── (additional raw files)
│   ├── processed/                   # Cleaned individual datasets
│   │   ├── *_clean.csv
│   │   ├── *_clean.csv
│   │   └── *_clean.csv
│   └── final/                       # Analysis-ready merged panel
│       ├── [dataset]_analysis_panel.csv  # Final merged dataset
│       └── data_dictionary.md            # Variable definitions
├── results/
│   ├── figures/
│   ├── reports/
│   └── tables/
├── tests/
│   └── .gitkeep
├── README.md                        # Team info, research question, dataset overview
├── M1_data_quality_report.md       # Data quality documentation
└── AI_AUDIT_APPENDIX.md            # AI disclosure (REQUIRED)
```

Data Pipeline Flow:

1. **Fetch scripts** (`fetch_*.py`) → Download/load data → Clean → Save to `data/processed/`
2. **Merge script** (`merge_final_panel.py`) → Combine processed datasets → Save to `data/final/`

Key Principles:

- One Python script per dataset (modular, easier to debug)
- Raw data stays untouched in `data/raw/`
- Each dataset gets cleaned independently in `data/processed/`
- Final merged panel in `data/final/` ready for M2 analysis

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 Scripts (Modular Pipeline)

You need multiple Python scripts, one per dataset plus one merge script:

A. Fetch Scripts (one per dataset)

Example: `code/fetch_[dataset1]_data.py`

```
"""
QM 2023 Capstone Project: M1 - [Dataset1] Data Fetch & Clean
Team: [Your Team Name]
Members: [List names]

This script fetches/loads primary dataset, cleans it, and saves to processed/.
"""

# Section 1: Imports and config_paths
# Section 2: Load raw data from data/raw/*_raw.csv
# Section 3: Clean missing values (drop/impute with justification)
# Section 4: Handle outliers (winsorize/cap)
# Section 5: Remove duplicates
# Section 6: Apply size/volume filters
# Section 7: Save to data/processed/*_clean.csv
```

Example: `code/fetch_[dataset2]_data.py`

```
"""
QM 2023 Capstone Project: M1 - [Dataset2] Data Fetch & Clean
Team: [Your Team Name]

This script fetches supplementary data (e.g., economic indicators via API), cleans it, and
saves to processed/.
"""

# Section 1: Imports and config_paths
# Section 2: Fetch data (API call or load CSV)
# Section 3: Align to appropriate time frequency
# Section 4: Handle missing values (forward fill, drop, interpolate)
# Section 5: Rename columns for consistency
# Section 6: Save to data/processed/*_clean.csv
```

Additional datasets: Create similar `fetch_[dataset].py` scripts for housing data, policy measures, etc.

B. Merge Script

`code/merge_final_panel.py`

```
"""
QM 2023 Capstone Project: M1 - Final Panel Merge
Team: [Your Team Name]

This script merges all processed datasets into final analysis panel.
"""

# Section 1: Imports and config_paths
# Section 2: Load all processed datasets from data/processed/*_clean.csv
# Section 3: Align time variables (convert to same format)
# Section 4: Merge datasets (left join on entity_id + time_var)
# Section 5: Verify merge integrity (row counts, no duplicates)
# Section 6: Save to data/final/[dataset]_analysis_panel.csv
# Section 7: Create data dictionary (data/final/data_dictionary.md)
```

Requirements for all scripts:

- ☑ Uses **relative paths only** (via `config_paths.py`)
- ☑ Runs without errors from top to bottom
- ☑ Clear section headers and comments
- ☑ Prints before/after row counts and summary statistics

---

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. Root README.md (Project Overview)

Location: Repository root (`README.md`)

Required sections:

```markdown
# QM 2023 Capstone Project: [Team Name]

## Team Members
- [Name 1] - [Role 1, e.g., Data Engineer]
- [Name 2] - [Role 2, e.g., Analyst]
- [Name 3] - [Role 3, e.g., Visualizer]
- [Name 4] - [Role 4, e.g., Writer]

## Research Question
[1-2 sentence clear research question. Example: "How do REIT returns respond to Federal
Reserve interest rate changes across different property sectors?"]

## Dataset Overview
- **Primary Dataset:** [Name, source, coverage]
  - Entities: [N] | Time: [frequency] | Period: [date range]
- **Supplementary Data:** [List 3-5 key supplementary variables]
  - FRED: FEDFUNDS, MORTGAGE30US, CPIAUCSL, UNRATE
  - [Other]: [Description]

## Hypotheses (Preliminary)
1. [Hypothesis 1]
2. [Hypothesis 2]
3. [Hypothesis 3]

## Repository Structure
[Copy the folder structure tree from above]

## How to Run
1. Clone repository
2. Open in GitHub Codespaces
3. Run fetch scripts: `python code/fetch_[dataset1]_data.py`, `python
code/fetch_[dataset2]_data.py`
4. Run merge script: `python code/merge_final_panel.py`
5. Check output: `data/final/[dataset]_analysis_panel.csv`
```

---

4. Data Dictionary: `data/final/data_dictionary.md`

Location: `data/final/data_dictionary.md`

Created automatically by `merge_final_panel.py`. Should contain:

- Dataset overview (N entities, N time periods, date range)
- Variable definitions table (variable, description, type, source, units)
- Cleaning decisions summary

See Section 5 ("Save Output") in Technical Requirements for template code.

---

5. 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.

---

6. 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 and Create Data Dictionary

Save merged panel:

```
merged.to_csv(FINAL_DATA_DIR / '[dataset]_analysis_panel.csv', index=False)
print(f"✓ Saved final panel: {merged.shape[0]} rows × {merged.shape[1]} columns")
```

Create data dictionary (markdown format):

```
# Create data_dictionary.md
data_dict = f"""# Data Dictionary: {dataset_name} Analysis Panel

## Dataset Overview
- **Dataset Name:** {dataset_name}
- **Number of Entities:** {merged['entity_id'].nunique()}
- **Number of Time Periods:** {merged['time_var'].nunique()}
- **Total Observations:** {merged.shape[0]}
- **Time Range:** {merged['time_var'].min()} to {merged['time_var'].max()}
- **Panel Structure:** {'Balanced' if balanced else 'Unbalanced'}

## Variable Definitions

| Variable | Description | Type | Source | Units |
|----------|-------------|------|--------|-------|
| permno | CRSP permanent identifier | int | REIT Master | ID |
| ym | Year-month | datetime | REIT Master | YYYY-MM |
| ret | Monthly total return | float | REIT Master | decimal (0.05 = 5%) |
| mcap | Market capitalization | float | REIT Master | millions USD |
| sector | REIT sector classification | str | REIT Master | category |
| fedfunds | Effective Federal Funds Rate | float | FRED | percent |
| mortgage30us | 30-Year Mortgage Rate | float | FRED | percent |
| cpiaucsl | Consumer Price Index | float | FRED | index (1982-84=100) |
| unrate | Unemployment Rate | float | FRED | percent |

## Cleaning Decisions Summary
- **Missing values:** Dropped 5.2% of observations with missing returns (delistings)
- **Outliers:** Winsorized returns at 1st/99th percentile
- **Size filter:** Dropped REITs with mcap < $10M
- **Duplicates:** Removed duplicate permno-month pairs (kept first)
"""

with open(FINAL_DATA_DIR / 'data_dictionary.md', 'w') as f:
    f.write(data_dict)
```

Grading Rubric (50 points)

Component	Points	Criteria

Component	Points	Criteria
Reproducibility	15	Scripts run; relative paths; modular structure; clear 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	README, data dictionary, data quality report complete; 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)

- [ ] All fetch scripts run without errors (`python code/fetch_*.py`)
- [ ] Merge script runs without errors (`python code/merge_final_panel.py`)
- [ ] Test from scratch: Delete `data/processed/` and `data/final/`, rerun all scripts
- [ ] No absolute paths (search for `C:\`, `/Users/` in all Python files)
- [ ] No manual Excel editing — scripts alone produce output
- [ ] Row counts: merged ≤ primary (no accidental duplication)
- [ ] Spot-check supplementary values (e.g., FEDFUNDS Jan 2020 ~1.55%)
- [ ] All deliverables present:
  - [ ] `README.md` (root) with team info and research question
  - [ ] Multiple `code/fetch_*.py` scripts
  - [ ] `code/merge_final_panel.py`
  - [ ] `data/final/*_analysis_panel.csv`
  - [ ] `data/final/data_dictionary.md`
  - [ ] `M1_data_quality_report.md`
  - [ ] `AI_AUDIT_APPENDIX.md`

- [ ] Team names in all documentation files
- [ ] Alternative dataset approved (if applicable) by Week 4

Submission Instructions

1. Verify all files are in the correct folders (see Repository Structure above)
2. Stage all M1 files in GitHub Codespaces:

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

3. Submit GitHub repository URL to Blackboard assignment
4. Verify all files visible on GitHub.com before submission deadline

Deadline: Wednesday, Feb 25, 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

QM 2023: Statistics II, Spring 2026, University of Tulsa | Dr. Cayman Seagraves (cayman-seagraves@utulsa.edu)