

# **Software, Rice Database, Technical Indicators**

BUSI 722: Data-Driven Finance II

---

Kerry Back

# Review: CAPM & Fama-French Factors

# The CAPM

The Capital Asset Pricing Model says that the expected excess return of any asset is proportional to the market's expected excess return:

$$\mathbb{E}[r_i] - r_f = \beta_i (\mathbb{E}[r_m] - r_f)$$

- $r_i$  = return on asset  $i$
- $r_f$  = risk-free rate
- $r_m$  = return on the market portfolio
- $\beta_i = \frac{\text{Cov}(r_i, r_m)}{\text{Var}(r_m)}$

## CAPM $\Leftrightarrow$ Zero Alphas

Run a time-series regression for each asset  $i$ :

$$r_{it} - r_{ft} = \alpha_i + \beta_i(r_{mt} - r_{ft}) + \varepsilon_{it}$$

- The CAPM is equivalent to saying  $\alpha_i = 0$  for every asset.
- If  $\alpha_i > 0$ : the asset earns more than the CAPM predicts — it is “underpriced.”
- If  $\alpha_i < 0$ : the asset earns less than the CAPM predicts — it is “overpriced.”

Testing the CAPM = testing whether alphas are jointly zero.

# The Size Effect

**Small stocks have historically earned higher average returns than large stocks**, even after adjusting for beta.

- Sort stocks into portfolios by market capitalization
- Small-stock portfolios earn positive CAPM alphas
- Large-stock portfolios earn negative CAPM alphas
- This is evidence against the CAPM

Size is measured by market capitalization = price  $\times$  shares outstanding.

**Value stocks (high book-to-market) have historically outperformed growth stocks (low book-to-market),** even after adjusting for beta.

- Book-to-market = book equity / market equity
- High B/M (“value”) stocks: positive CAPM alphas
- Low B/M (“growth”) stocks: negative CAPM alphas
- This is further evidence against the CAPM

# Fama-French Three-Factor Model (1993)

Fama and French proposed replacing the CAPM with a three-factor model:

$$\mathbb{E}[r_i] - r_f = \beta_i^{\text{MKT}} \cdot \text{MKT} + \beta_i^{\text{SMB}} \cdot \text{SMB} + \beta_i^{\text{HML}} \cdot \text{HML}$$

- **MKT** =  $r_m - r_f$ : market excess return
- **SMB** (Small Minus Big): return on small stocks minus return on large stocks
- **HML** (High Minus Low): return on high B/M stocks minus return on low B/M stocks

# Interpreting the Three-Factor Model

The three-factor model says that expected returns are explained by exposures to three sources of risk.

- The model “works” if alphas are zero when we regress excess returns on the three factors:

$$r_{it} - r_{ft} = \alpha_i + \beta_i^{\text{MKT}} \text{MKT}_t + \beta_i^{\text{SMB}} \text{SMB}_t + \beta_i^{\text{HML}} \text{HML}_t + \varepsilon_{it}$$

- The size and value effects that produced nonzero CAPM alphas are captured by SMB and HML.
- But ... other anomalies (momentum, profitability, investment) still produce nonzero alphas.

# Setup: Claude Pro & Claude Code

# Initial Steps

## 1. Sign up for Claude Pro:

1. Visit [claude.ai](https://claude.ai)
2. Click “Sign Up” or “Get Started”
3. Create account using email or Google/Apple sign-in
4. After signing in, click “Upgrade to Pro” in the sidebar
5. Choose the monthly payment plan
6. Complete payment information

## 2. [YouTube Video](#)

# Mac – Install Claude Code

## Step 1: Install Node.js

- Using Homebrew: `brew install node`
- Or download installer from [nodejs.org](https://nodejs.org)

## Step 2: Install Claude Code

- Run: `npm install -g @anthropic-ai/claude-code`

## Step 3: Verify Installation

- Run: `claude doctor`

# Windows – Install Claude Code

## Step 1: Install Node.js

- Download and install from [nodejs.org](https://nodejs.org)
- Choose the LTS (Long Term Support) version

## Step 2: Install Git for Windows

- Download from [git-scm.com/download/win](https://git-scm.com/download/win)
- During installation, select “Git Bash” (default option)

## Step 3: Install Claude Code

- In PowerShell, run: `npm install -g @anthropic-ai/claude-code`

# Install Python & Create Virtual Environment

## Install Python:

- Tell Claude Code: “Install Python 3.13 and add it to the path.”
- Tell Claude Code: “Upgrade pip”

## Create Virtual Environment:

- Tell Claude Code: “Create a virtual environment using Python 3.13 in my current directory.”

## Install packages in the virtual environment:

- `numpy pandas scipy statsmodels scikit-learn`

# VS Code & Database Setup

# Install and Open VS Code

- Install VS Code: [code.visualstudio.com](https://code.visualstudio.com)
- Launch VS Code
- File → Open Folder → navigate to your course folder
- Install extensions: Python, Jupyter, Claude Code, Data Wrangler, Rainbow CSV
- View → Command Palette → “Python: Select Interpreter” → choose venv

## Launch Claude Code in VS Code

- If you see the orange Claude icon in the top toolbar, click it.
- If not, create a new file (File → New Text File) and you should see it.
- Or View → Command Palette and enter “Claude Code: Open in New Tab.”

Test: Ask Claude Code: What is the sum of the first 1,000 integers?

- Database Guide: <https://portal-guide.rice-business.org>
- Visit [data-portal.rice-business.org](https://data-portal.rice-business.org) to get an access token
- The data portal is an AI agent that uses ChatGPT to generate SQL and query the database

## Create .env File:

- Tell Claude Code to create a .env file
- Add: `RICE_ACCESS_TOKEN=your_token_here`

Skills are text files that give Claude Code specialized knowledge.

The easiest way to install is to **ask Claude Code to do it for you**: “Download and install the rice-data-query and merge skills, plus CLAUDE.md from [mgmt638.kerryback.com/skills/](https://mgmt638.kerryback.com/skills/)”

## Workflow

Visit the Data Guide to find variable names from the SF1 (Sharadar Fundamentals) table.

Tell Claude Code to:

1. Get returns (monthly or weekly) from the Rice database.
2. Get the variables you want from SF1.
3. Calculate any ratios or growth rates you want.
4. Merge the returns with the fundamental data.

Save the merged data. **You do not need to repeat this process if you already have the data you want.**

## Timing of Data

During merging, Claude will align variables so that

- Fundamental variables are all known at the beginning of the period
- Momentum is momentum as of the beginning of the period
- Close is the closing price at the end of the previous month or week
- Return is the return over the period (from beginning to end)

All variables other than return **are known at the beginning of the period** and can be used to pick stocks.

## Avoiding Look-Ahead Bias

- Price data (close, marketcap, pb) lagged by 1 month
- Fundamentals available in first full month *after* filing date
- Forward-filled until next filing
- Include **all tickers** (including delisted) to avoid survivorship bias

Claude should automatically save data as parquet files. This is a compact, fast format.

To view the data, ask Claude to:

- **“Convert the file to Excel.”** Then open as usual.
- **“Convert the file to csv.”** Then double-click in VS Code File Explorer.
- **“Read the data in a Jupyter notebook.”** Then work with the data in Python.