Lesson 9

Working with CRSP U.S. Daily Stock Data

Data Fields, Coding Schemes, and Python Tools

Christopher Ting

✉: **cting@hiroshima-u.ac.jp**

☺: **https://cting.neocities.org/**

Reference: *CRSP Data Description Guide for CRSPAccess (FIZ) US Stock & Index Databases*, July 2025

Outline

1. Introduction to CRSP
2. Our Local Dataset
3. Data Dictionary: Header & Name Identification
4. Data Dictionary: Distributions, Shares & Delisting
5. Data Dictionary: NASDAQ, Price/Volume/Return & Benchmarks
6. Coding Schemes
7. Key Conventions & Calculations
8. Data Wrangling in Python
9. Python Programs for Plotting
10. Wrap-Up

Outline

1. Introduction to CRSP
2. Our Local Dataset
3. Data Dictionary: Header & Name Identification
4. Data Dictionary: Distributions, Shares & Delisting
5. Data Dictionary: NASDAQ, Price/Volume/Return & Benchmarks
6. Coding Schemes
7. Key Conventions & Calculations
8. Data Wrangling in Python
9. Python Programs for Plotting
10. Wrap-Up

What is CRSP?

- **CRSP** = Center for Research in Security Prices, a research institute of the University of Chicago Booth School of Business (Prof. Eugene Fama, Chairman).
- Founded 1960 after an inquiry from Merrill Lynch's Louis Engel to Prof. James Lorie; original design and content built by Lawrence Fisher and James Lorie.
- Produces the academic "research-grade" database of US security prices, dividends, and returns used throughout empirical finance.
- Files provided: common stocks (NYSE, NYSE MKT/AMEX, NASDAQ, Arca, Cboe BZX); CRSP indices; NASDAQ and S&P 500 composite indices; industry indices; US Treasury bonds; survivor-bias-free mutual funds; and more.

Data Coverage by Exchange

Exchange	Monthly file begins	Daily file begins
NYSE	12/31/1925	12/31/1925
NYSE MKT (AMEX)	07/31/1962	07/02/1962
NASDAQ	12/29/1972	12/14/1972
NYSE Arca	03/31/2006	03/08/2006
Cboe BZX	01/31/2012	01/24/2012

- In 2005 CRSP completed merging 1925–1962 daily data for NYSE securities, giving seamless daily coverage back to December 1925.
- NASDAQ closing bid/ask and number-of-trades data begin 11/1/1982 (National Market) and 6/15/1992 (SmallCap Market).

CRSP Calendars

- A CRSP *calendar* is the set of time periods a series is indexed to. The **daily calendar** contains only dates on which a major US exchange actually traded — no weekends, no holidays.
- Every date field in the files (`date`, `DCLRDT`, `PAYDT`, `RCRDDT`, ...) is an integer in **YYYYMMDD** format.
- Other calendars (monthly, weekly, quarterly, annual) are derived from the daily calendar by taking its last trading date in each period.

Outline

1. Introduction to CRSP
2. Our Local Dataset
3. Data Dictionary: Header & Name Identification
4. Data Dictionary: Distributions, Shares & Delisting
5. Data Dictionary: NASDAQ, Price/Volume/Return & Benchmarks
6. Coding Schemes
7. Key Conventions & Calculations
8. Data Wrangling in Python
9. Python Programs for Plotting
10. Wrap-Up

Directory Layout: C:\CRSP_daily

```
CRSP_daily/
  header.csv          <- the 63 column names
  header.txt          <- same, human readable
  listofstocks.txt    <- every PERMNO in the
      set
  datalist.txt
  *_Data_Descriptions_Guide.pdf  (2x)
  plot_price.py
  plot_volume.py
  plot_returns.py
  plot_bidask.py
  USStocks/
    10000.csv
    10001.csv
    ...
    (26,473 files)
```

- **One CSV per security** in `USStocks/`, named `<PERMNO>.csv`.
- Every file shares the same 63-column header.
- `listofstocks.txt` enumerates the PERMNOs available so you can loop over the whole universe.
- Two vendor PDFs are the authoritative field/code reference; this deck distills them.

Anatomy of One File

- Each `USStocks/<PERMNO>.csv` is **one row per trading day** for that security, sorted by `date`.
- The very first row is often a "header-only" row: identifiers are populated but there is no trade yet.
- A handful of dates can appear twice (duplicate rows) — always `drop_duplicates(subset='date')` before analysis.

Example — first two rows of `10000.csv`

```
PERMNO,date,...,BIDLO,ASKHI,PRC,VOL,RET,BID,ASK,SHROUT,CFACPR,...
10000,19860106,...,,,,,,,,,,,...
10000,19860107,...,2.375,2.75,-2.5625,1000.0,C,,,3680.0,1.0,...
```

Negative `PRC` and the letter code in `RET` are both normal — explained later.

PERMNO and PERMCO: the Permanent Keys

PERMNO — Permanent security number

A unique ID CRSP assigns to each *security*. It never changes and is never reused, even through ticker changes, name changes, or CUSIP changes. This is the primary key of the daily file and the filename of every CSV in `USStocks/`.

PERMCO — Permanent company number

A unique ID CRSP assigns to each *company*. It stays constant across all of a company's securities (e.g. multiple share classes) and across the company's own name/capital-structure changes.

Rule of thumb

Track one security's full history with `PERMNO`; aggregate multiple share classes of the same firm with `PERMCO`.

Outline

1. Introduction to CRSP
2. Our Local Dataset
3. Data Dictionary: Header & Name Identification
4. Data Dictionary: Distributions, Shares & Delisting
5. Data Dictionary: NASDAQ, Price/Volume/Return & Benchmarks
6. Coding Schemes
7. Key Conventions & Calculations
8. Data Wrangling in Python
9. Python Programs for Plotting
10. Wrap-Up

The 63 Fields, Organized

CRSP's own file structure groups the columns into arrays. We will work through the data dictionary in that order — it is the most useful way to remember what belongs together.

1. Header Identification (7 fields) – static per security
2. Name History (13 fields) – changes when name/exchange/ticker/CUSIP changes
3. Distribution Events (9 fields) – one event per dividend/corporate action
4. Shares Outstanding (3 fields)
5. Delisting Event (8 fields)
6. NASDAQ Information (4 fields)
7. Price / Volume / Return core series (5 fields)
8. Auxiliary daily series (5 fields)
9. Split/dividend adjustment factors (2 fields)
10. Appended market-index benchmark returns (5 fields)
11. `date` itself (1 field)

$7 + 13 + 9 + 3 + 8 + 4 + 5 + 5 + 2 + 5 + 1 = 63$

Header Identification (static per security)

Field	Description
PERMNO	Permanent security identifier (see previous slide).
PERMCO	Permanent company identifier (see previous slide).
ISSUNO	NASDAQ Issue Number assigned by NASD; distinguishes multiple issues of one company on NASDAQ. 0 if never NASDAQ-listed.
HEXCD	Header Exchange Code — most recently known exchange (numeric code, see Coding Schemes).
HSICCD	Header SIC Code — last non-zero SIC code on file.
HSICMG	Header SIC *Major Group*: first 2 digits of HSICCD.
HSICIG	Header SIC *Industry Group*: first 3 digits of HSICCD.

SIC hierarchy: 2 digits = major group, 3 digits = industry group, 4 digits = full industry. Note: our file also carries a plain CUSIP column, covered on the Name History slide since in practice it is populated per name record.

Name History (I): Identifiers

A new name-history record is added whenever CUSIP, company name, exchange, ticker, share class, or SIC code changes; the fields below describe *that* record.

Field	Description
`NAMEENDT`	Last date (YYYYMMDD) this name record is in effect.
`CUSIP`	8-character CUSIP identifier tied to the header record.
`NCUSIP`	8-character CUSIP effective during *this* name record.
`TICKER`	Ticker symbol for this name record.
`TSYMBOL`	Trading ticker symbol actually used for quotation/trading (can differ slightly from `TICKER`, e.g. class suffixes).
`COMNAM`	Company name for this name record.
`SHRCLS`	Share class letter (e.g. “A”); usually blank.

Name History (II): Classification & Status

Field	Description
SICCD	SIC code effective for this name record.
NAICS	North American Industry Classification System code (introduced 1997 as SIC's successor; populated from 2004 onward).
PRIMEXCH	Primary listing exchange, one letter: N/A/Q/R/B/I/X (see Coding Schemes).
EXCHCD	Numeric exchange code for this period (see Coding Schemes); includes special negative codes for halted/suspended issues.
TRDSTAT	Trading status, one letter: A/H/S/X.
SECSTAT	Security status, one letter: W/R/E/Q/X.
SHRCD	Share Code — 2-digit Share Type code (see Coding Schemes). SHRCD $\in \{10, 11\}$ is the standard "ordinary common stock" filter.

Outline

1. Introduction to CRSP
2. Our Local Dataset
3. Data Dictionary: Header & Name Identification
4. Data Dictionary: Distributions, Shares & Delisting
5. Data Dictionary: NASDAQ, Price/Volume/Return & Benchmarks
6. Coding Schemes
7. Key Conventions & Calculations
8. Data Wrangling in Python
9. Python Programs for Plotting
10. Wrap-Up

Distribution Events (I): the Corporate Action Itself

Each event in a security's distribution history — cash dividend, split, spin-off, merger, rights offering, . . . — is coded onto the daily row for its Ex-Distribution Date.

Field	Description
`DISTCD`	4-digit Distribution Code: event type / payment method / event detail / tax status (full structure in Coding Schemes).
`DIVAMT`	Dividend Cash Amount, US$ per share.
`FACPR`	Factor to Adjust Price (see Key Conventions for the formula).
`FACSHR`	Factor to Adjust Shares Outstanding (parallel to `FACPR`; equal to it for ordinary splits/dividends, can differ for spin-offs/rights).

Distribution Events (II): Dates and Linked Securities

Field	Description
DCLRDT	Distribution Declaration Date — when the board declared it.
RCRDDT	Record Date.
PAYDT	Payment Date — when checks were mailed / distribution made.
ACPERM	Acquiring PERMNO — the security a shareholder received stock in (spin-off, exchange, merger). < 1000 if not applicable.
ACCOMP	Acquiring PERMCO — the company counterpart of ACPERM.

Both ACPERM/ACCOMP are incomplete prior to 1985.

Shares Outstanding

Field	Description
SHROUT	Shares outstanding, in **thousands**, *not* adjusted for splits. Company-level treasury shares excluded; for ADRs this is the ADR's own share count, not the underlying issue's.
SHRFLG	Shares Outstanding Observation Flag — source/quality flag for the SHROUT observation.
SHRENDDT	Shares Outstanding Observation End Date — last date this SHROUT observation is valid.

Multiply by CFACSHR to get a split-adjusted, comparable share count over time (see Key Conventions).

Delisting Event

Every security has exactly one delisting record; for an active issue, the delisting date is simply the last date of available price data.

Field	Description
DLSTCD	Delisting Code (3-digit; full scheme in Coding Schemes).
DLAMT	Amount After Delisting: off-exchange price/quote, or sum of post-delisting distribution payments. 0 if still active or unknown.
DLPRC	Delisting Price: positive = trade price, negative = bid/ask average, 0 = active or unavailable.
DLRET / DLRETX	Delisting Return, with / without dividends paid between the last trade and the delisting payment date.
NEXTDT	Delisting Date of Next Available (price) Information.
DLPDT	Delisting Payment Date.
NWPERM	New PERMNO — successor security that continues a shareholder's position after a merger/exchange.

Outline

1. Introduction to CRSP
2. Our Local Dataset
3. Data Dictionary: Header & Name Identification
4. Data Dictionary: Distributions, Shares & Delisting
5. Data Dictionary: NASDAQ, Price/Volume/Return & Benchmarks
6. Coding Schemes
7. Key Conventions & Calculations
8. Data Wrangling in Python
9. Python Programs for Plotting
10. Wrap-Up

NASDAQ Information (NASDAQ-listed issues, from Nov. 1982)

Field	Description
MMCNT	Market Maker Count for the issue.
NSDINX	NASD Index Code — the issue's internal NASD industry classification (INDS/BANK/INSR/. . . ; see Coding Schemes).
NMSIND	NASDAQ National Market Indicator (0–3; see Coding Schemes).
TRTSCD	NASDAQ Traits Code.

Data are missing for all issues in February 1986, and for NASD company numbers below 1025 in December 1982.

Price / Volume / Return — the Daily Core

Field	Description
BIDLO	Low trade price for the day (or bid low if no trade).
ASKHI	High trade price for the day (or ask high if no trade).
PRC	Closing price; **negative** value = closing bid/ask average substituted because no trade price was available.
VOL	Shares traded that day, unadjusted for splits.
RET	Holding-period total return (dividends reinvested on the ex-date). Occasionally a one-letter text code instead of a number (see Coding Schemes / Data Wrangling).

Highs/lows/volumes are consolidated across all US exchanges the security traded on that day, not just its primary listing.

Auxiliary Daily Series

Field	Description
RETX	Return *without* dividends — capital appreciation (price) return only.
OPENPRC	Opening price. Sparse/blank for most of history; reliably populated for Arca-era and recent data.
BID	Closing bid quote.
ASK	Closing ask quote.
NUMTRD	Number of trades that day (NASDAQ issues only).

BID/ASK coverage typically starts well after a security's price history begins — see the Data Wrangling section for a real example.

Split/Dividend Adjustment Factors

Field	Description
CFACPR	Cumulative Factor to Adjust **Price**: divide PRC (or BID/ASK/BIDLO/ASKHI) by this to get a price comparable across the security's entire split history.
CFACSHR	Cumulative Factor to Adjust **Shares/Volume**: multiply SHROUT or VOL by this for the same reason.

Both factors are indexed near 1.0 at the most recent date and grow *larger* further back in time, in proportion to the cumulative split ratio since then. Formulas and worked examples are in *Key Conventions*.

Appended Market-Index Benchmark Returns

These five columns are **not** part of a security's own CRSPAccess record — they are that day's CRSP/S&P market-index return, merged onto every security's row so you always have a benchmark alongside the stock.

Field	Description
`vwretd`	CRSP value-weighted market return (NYSE/AMEX/NASDAQ), *including* distributions.
`vwretx`	Same universe, value-weighted, price return only (excludes distributions).
`ewretd`	CRSP equal-weighted market return, including distributions.
`ewretx`	CRSP equal-weighted market return, price return only.
`sprtrn`	S&P 500 Composite return.

date

The calendar trading date, an integer in **YYYYMMDD** format, keyed to CRSP's daily trading calendar. Weekends and exchange holidays never appear.

In pandas:

```
pd.to_datetime(stock.date, format='%Y%m%d')
```

Outline

1. Introduction to CRSP
2. Our Local Dataset
3. Data Dictionary: Header & Name Identification
4. Data Dictionary: Distributions, Shares & Delisting
5. Data Dictionary: NASDAQ, Price/Volume/Return & Benchmarks
6. Coding Schemes
7. Key Conventions & Calculations
8. Data Wrangling in Python
9. Python Programs for Plotting
10. Wrap-Up

SHRCD: Share Type — First Digit

SHRCD is a 2-digit code: 10 × (first digit) + (second digit).

Code	Security type
1	Ordinary common shares
2	Certificates, Americus Trust Components (Prime, Score, & Units)
3	ADRs (American Depositary Receipts)
4	SBIs (Shares of Beneficial Interest)
7	Units (depositary units, limited-partnership units, . . .), Exchange Traded Funds

Codes 3 and 6 are not assigned; 2+“3” is reserved for Americus Trust Components and 7+“3” for ETFs.

SHRCD: Share Type — Second Digit

Code	Refinement
0	Securities which have not been further defined
1	Securities which need not be further defined
2	Companies incorporated outside the US
3	Americus Trust Components / Exchange Traded Funds
4	Closed-end funds and unit investment trusts
5	Closed-end fund companies incorporated outside the US
8	REITs (Real Estate Investment Trusts)

Example: SHRCD=11 $\rightarrow$ ordinary common share, needs not be further defined — the most common code for a plain US common stock. $\{10, 11\}$ is the standard "common stock only" filter.

EXCHCD / HEXCD: Exchange & Index Codes

Code	Exchange	Code	Exchange
−2	Halted by NYSE or NYSE MKT	4	Arca
−1	Suspended by NYSE/NYSE MKT/NASDAQ	5	Cboe BZX (as quoted by NASDAQ)
0	Not trading on NYSE/NYSE MKT/NASDAQ	6	IEX
1	NYSE	10	Boston Stock Exchange
2	NYSE MKT (AMEX)	13	Chicago Stock Exchange
3	NASDAQ	16	Pacific Stock Exchange
17	Philadelphia Stock Exchange	19	Toronto Stock Exchange
20	Over-the-Counter (non-NASDAQ)	31	When-issued trading on NYSE
32	When-issued trading on NYSE MKT	33	When-issued trading on NASDAQ
34	When-issued trading on Arca		

This is the scheme for both EXCHCD (period value) and HEXCD (header/most-recent value).

PRIMEXCH, TRDSTAT, SECSTAT: Letter Codes

PRIMEXCH

Code	Exchange
N	NYSE
A	NYSE MKT
Q	NASDAQ
R	Arca
B	Cboe BZX
I	IEX
X	Other

TRDSTAT

Code	Status
A	Active
H	Halted
S	Suspended
X	Unknown

SECSTAT

Code	Status
W	When issued
R	Regular way
E	Ex-distributed
Q	Non-leading WI
X	Untracked / unknown

Example from the sample row on the “Anatomy” slide: PRIMEXCH=Q (NASDAQ), TRDSTAT=A (active), SECSTAT=R (regular way).

DISTCD: Distribution Code Structure

A 4-digit code: *[event type][payment method][event detail][tax status]*.

1st digit: Event type		2nd digit: Payment method	
1	Ordinary dividend	0	Unknown, not yet coded
2	Liquidating dividend	1	Unspecified or not applicable
3	Exchanges and reorganizations	2	Cash, US dollars
4	Subscription rights	3	Cash, foreign currency converted to USD
5	Splits and stock dividends	5	Same issue of common stock
6	Notation of issuance (shares change)	6	Units incl. same issue of common
7	General announcement, dropped issues	7	A different common issue on file
		8	Other property

4th digit: Tax status (0/1 = unknown/n.a. as above)			
2 Normal, taxable same rate as dividends taxable as ordinary income	3: Normal, non-taxable	4: Return of capital	8: Fully

DISTCD: 3rd Digit Depends on the 1st Digit

1st digit	3rd digit meaning
1 (dividend)	Frequency: 2 monthly, 3 quarterly, 4 semi-annual, 5 annual, 6 year-end, 7 extra/special, 8 interim, 9 non-recurring
2 (liquidation)	Event: 3 partial, 4 step, 5 final, 6 approval, 7 sale of assets, 8 court proceeding
3 (exchange/reorg)	Event: 2 merger, 5 non-ordinary distribution, 6 reorg, 7 option of stock, 8 exchange
4 (rights)	Valuation method: 2 market value at exdate, 3 fair market value, 4 value at exdate (calc.), 5–7 non-transferable variants
5 (split)	Type: 2 split, 3 stock dividend, 4 split & stock dividend, 5 option of cash, 6/7 distribution of a different issue
6 (issuance)	Reason: 2 merger (on file), 3 merger (not on file), 4 stock conversion, 6 buy-back, 7 exchange offer, 8 offering
7 (dropped)	Reason: 1 bankruptcy, 2 negative performance, 3–4 failed tender offer, 5–6 govt./external intervention, 8 exchange requirement failure

Example: `5523` = split (5), cash (2, unused for splits) . . . actually `5223` reads as split/2/split-type 2/non-taxable.

DISTCD: Common Examples

Code	Meaning
1200	US cash dividend, tax status unknown
1232	US cash dividend, quarterly, taxable same rate as dividends
1252	US cash dividend, annual, taxable same rate as dividends
1272	US cash dividend, extra or special, taxable same rate as dividends
2255	Cash paid as a final liquidating payment
3222	Cash received in a merger, taxable same rate as dividends
3763	Received as a spin-off in reorganization, non-taxable
4523	Rights to buy more of this security, non-taxable
5523	**Stock split, non-taxable**
5533	Stock dividend, non-taxable
6511	Common shares increased/decreased, reason unspecified

Abbreviated; the guide's "Commonly Coded Distribution Events" table lists roughly 90 curated examples across dividends, liquidations, mergers, rights, splits, and share-count changes.

DLSTCD: Delisting Code Categories

A 3-digit code, grouped by first digit:

Range	Category
100	Active / still trading
200–290	Mergers
300–390	Exchanges
400–490	Liquidations
500–591	Dropped by exchange (delisted for cause)
600–610	Expirations (warrants/rights/units)
900–903	Domestic security becomes foreign

The next slides give the most commonly used individual codes in each range.

DLSTCD: Active, Mergers, Exchanges (selected)

Code	Description	Code	Description
100	Still trading NYSE/NYSE MKT/NASDAQ/Arca/Cboe BZX/IEX	231	Merged: shareholders receive common stock/ADRs
150	Active, no prices in this file version	233	Merged: shareholders receive cash
170	Stopped trading, not formally delisted	241	Merged: common stock + cash, issue on file
200	Acquired in merger, payment details unknown	280	Delisted due to failed merger attempt
201–203	Merged into issue on NYSE/NYSE MKT/NASDAQ (retired code)	300	Acquired by exchange of stock, details unknown
205	Shareholders primarily receive mutual fund shares	331	Exchanged, primarily for another class of common
		333	Exchanged, primarily for cash
		351–353	Exchanged for stock/cash/property, issue not on CRSP file

DLSTCD: Liquidations & Dropped (selected)

Code	Description	Code	Description
400	Stopped trading due to liquidation	550–552	Delisted: too few market makers / shareholders / low price
450	Liquidated, final distribution verified	560–561	Delisted: insufficient capital or float
460–470	Liquidated, distribution unverified / pending	570–575	Delisted by company request (various reasons)
490	Liquidated, no distributions to be paid	580–587	Delisted: filing/registration/governance violations
500	Stopped trading, reason unavailable	591	Delisted: required by the SEC
501–520	Moved to another exchange or OTC	600–610	Warrant/right/unit expired, called, or split into parts
535	Delisted: unlisted trading privileges	900–903	Domestic security becomes foreign (may keep trading)

Full enumeration (about 90 codes total) is in Chapter 5 of the guide PDF.

NASDAQ Codes: Status & National Market Indicator

NASDAQ Status Code

Code	Meaning
0	Unknown or n/a
1	Active
2	Trading, one market maker
3	Suspended
4	Inactive
5	Delisted

`NMSIND` — National Market Indicator

Code	Meaning
0	Unknown or unavailable
1	SmallCap Market, before 6/15/1992
2	National Market
3	SmallCap Market, after 6/15/1992

NSDINX: NASD Index Code (excerpt)

Code	Category	Code	Category
1	NONE – no index	8	IXTC – Telecommunication
2	INDS – Industrial	9	IXCO – Computer
3	BANK – Bank	10	IXBT – Biotechnology
4	OFIN – Other Finance	11	IXCM – Composite
5	INSR – Insurance	12	NDX – Nasdaq 100
6	TRAN – Transportation	18	IXHC – Health Care
7	IXUT – Utility	41	RCMP – Cap Market Composite

This field has roughly 90 assigned codes in total (Chapter 5 of the guide); the ones above cover the broad sector groupings you will see most often.

Missing-Value Codes for RET/DLRET

The internal CRSPAccess API represents a missing return as one of these numeric sentinels:

Value	Reason
−44.0	Missing excess return: no portfolio assignment
−55.0	Missing delisting return (pending or unresolved)
−66.0	Gap of more than 10 trading days to the latest preceding price
−77.0	Not trading on an included exchange this file
−88.0	Out of range of the security's Begin/End data indices
−99.0	Missing return because the price itself is missing

In the CSV files you actually have

A small number of rows carry a **single letter** (e.g. `'C'`) in RET/RETX instead of a number or one of the codes above — typically the first trading day after a listing gap, where there is no valid prior price to compute a return from. Read these columns as text and handle the letters explicitly; see *Data Wrangling*.

Outline

1. Introduction to CRSP
2. Our Local Dataset
3. Data Dictionary: Header & Name Identification
4. Data Dictionary: Distributions, Shares & Delisting
5. Data Dictionary: NASDAQ, Price/Volume/Return & Benchmarks
6. Coding Schemes
7. Key Conventions & Calculations
8. Data Wrangling in Python
9. Python Programs for Plotting
10. Wrap-Up

Convention: Negative PRC

- If no trade price is available for a day, CRSP substitutes the **bid/ask average** into PRC and stores it as a **negative** number.
- This is a flag, not a real negative price — always take abs(PRC) before using it.
- Before 11/1/1982 (National Market) or 6/15/1992 (SmallCap), NASDAQ PRC is *always* a negative bid/ask average, since NASDAQ did not report trade prices before those dates.

In pandas

```
price = stock.PRC.abs() / stock.CFACPR
```

Convention: Adjusting for Splits

$$\text{adjusted price}_t = \frac{\texttt{PRC}_t}{\texttt{CFACPR}_t} \qquad \text{adjusted shares}_t = \texttt{SHROUT}_t \times \texttt{CFACSHR}_t$$

$$\text{adjusted volume}_t = \texttt{VOL}_t \times \texttt{CFACSHR}_t$$

For an ordinary split/stock dividend, FACPR (and FACSHR) on the ex-date is

$$\texttt{FACPR} = \frac{s(t) - s(t')}{s(t')}$$

where $s(t)$ is shares outstanding just after the split and $s(t')$ just before (t' = last date before the split). A 2-for-1 split gives FACPR$= 1$; a reverse split gives a value between -1 and 0.

- Cash dividends: FACPR$= 0$ (price factor doesn't move).
- Mergers / total liquidation / issue disappears: FACPR$= -1$ by convention.
- Spin-offs / rights: FACPR $=$ DIVAMT / price on the ex-date.

Convention: RET vs. RETX

RET — total return

Daily change in total investment value, **with** ordinary dividends reinvested on the ex-date.

RETX — price return

Daily change in price only, dividends **excluded**.

Use RET for total-return analysis (performance, cumulative growth of $1); use RETX when you specifically want price appreciation, e.g. as a fallback multiplier for filling a short bid/ask gap (a day's |RETX| approximates how far quotes moved).

Convention: How Delisting Return Is Computed

For an issue closed to further research, CRSP applies these rules in order:

1. If a price within 10 trading periods of the delist date exists, use it.
2. Else if a final distribution is known, use all distributions after the last available price.
3. Else if some (non-final) distributions are known, treat them as if they were the final distribution.
4. If there is evidence the stock is worthless, $\texttt{DLRET} = -1$ (a 100% loss).

If none of these apply and the issue is still pending research, `DLRET` is set to the missing code -55.0. `DLRETX` follows the same logic but excludes ordinary dividends paid between the last trade and the delisting payment date.

Outline

1. Introduction to CRSP
2. Our Local Dataset
3. Data Dictionary: Header & Name Identification
4. Data Dictionary: Distributions, Shares & Delisting
5. Data Dictionary: NASDAQ, Price/Volume/Return & Benchmarks
6. Coding Schemes
7. Key Conventions & Calculations
8. Data Wrangling in Python
9. Python Programs for Plotting
10. Wrap-Up

Reading a CSV Robustly

`RET`/`RETX` must be read as text — they occasionally hold a letter code, which would otherwise crash a numeric parse.

```
import pandas as pd

datadir = 'C:/CRSP_daily/USStocks/'
permno = 10866
stock = pd.read_csv(datadir + f'{permno}.csv',
    usecols=['date', 'COMNAM', 'TICKER', 'PRIMEXCH',
             'OPENPRC', 'BIDLO', 'ASKHI', 'PRC', 'VOL',
             'CFACPR', 'CFACSHR', 'RETX', 'RET', 'BID', 'ASK'],
    converters={'RETX': str, 'RET': str})

stock.drop_duplicates(subset='date', keep='first',
                      ignore_index=True, inplace=True)
```

Duplicate Dates and No-Trade Days

A blank `RETX` means CRSP has no trade to report that day (a halt, or a very illiquid stock); treat it as flat price and zero volume, carried from the prior row. A one-letter `RETX` means a return genuinely could not be computed — leave it alone.

```
for i in range(len(stock)):
    if stock.RETX[i].isalpha():
        continue                          # e.g. 'C' -- leave as is
    if stock.RETX[i] == '':
        stock.at[i, 'RETX'] = '0'
        stock.at[i, 'RET']  = '0'
        stock.at[i, 'VOL']  = 0
        for col in ['BIDLO', 'ASKHI', 'BID', 'ASK', 'PRC', 'OPENPRC']:
            stock.at[i, col] = stock[col][i - 1]
```

Real Example: a 40-Year Gap in BID/ASK

- PERMNO 10145 (Honeywell) has daily prices back to 1925 – but `BID/ASK` are missing for **10,311 consecutive rows**, 1952-01-28 through 1992-12-24.
- That is not a data error to "fill in" — bid/ask quotes for many older NYSE issues simply were not captured that far back.
- **Lesson:** distinguish a short, fillable hole (1–2 days, safe to carry the previous quote forward) from a genuine multi-year coverage gap (skip it; plot only the longest trustworthy stretch).

Naive carry-forward would have silently faked 40 years of quotes

Carrying the last 1952 quote forward day by day, scaled by $|\texttt{RETX}|$, produces a smoothly evolving but entirely fictitious 40-year midquote series. Always check the length of a gap before bridging it.

Outline

1. Introduction to CRSP
2. Our Local Dataset
3. Data Dictionary: Header & Name Identification
4. Data Dictionary: Distributions, Shares & Delisting
5. Data Dictionary: NASDAQ, Price/Volume/Return & Benchmarks
6. Coding Schemes
7. Key Conventions & Calculations
8. Data Wrangling in Python
9. Python Programs for Plotting
10. Wrap-Up

Four Programs, One Pattern

All four scripts live next to this deck in `C:\CRSP_daily` and share the same shape:

1. Load one `PERMNO`'s CSV, drop duplicate dates.
2. Clean/adjust the series being plotted.
3. Draw the chart and save a PNG named `<permno>_<series>.png`.

Script	Plots
`plot_price.py`	Split-adjusted closing price
`plot_volume.py`	Split-adjusted daily volume
`plot_returns.py`	Daily & cumulative return vs. `vwretd`
`plot_bidask.py`	Split-adjusted bid/ask midquote

Run from a terminal: `python plot_price.py 10145` (PERMNO defaults if omitted).

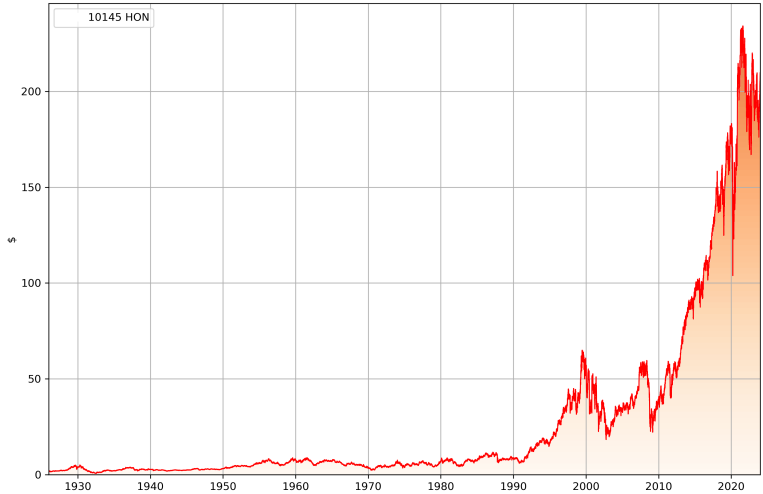
plot_price.py: Core Logic

```
# PRC is negative on bid/ask-average days -- take abs() before adjusting
stock['adj_price'] = stock.PRC.abs() / stock.CFACPR

xdata = np.array(pd.to_datetime(stock.date, format='%Y%m%d'))
plot_gradient_series(xdata, stock.adj_price.to_numpy(), '$',
                     f'{permno} {stock.TICKER.iloc[-1]}',
                     f'{permno}_price.png')
```

The gradient-filled area chart (‘plot_gradient_series’) paints an orange gradient behind the price line and whites out everything above it, so the price reads as a filled “water level” rather than a bare line.

Output: `plot_price.py` on PERMNO 10145 (Honeywell)



plot_bidask.py: Finding a Trustworthy Segment

```python
def longest_fillable_segment(good, max_gap):
    """Longest run of quotes whose internal holes are all <= max_gap days."""
    segments, seg_start, i, n = [], None, 0, len(good)
    while i < n:
        if good[i]:
            seg_start = seg_start if seg_start is not None else i
            i += 1
            continue
        j = i
        while j < n and not good[j]:
            j += 1
        if seg_start is not None and (j - i) <= max_gap and j < n:
            i = j                          # bridge a short internal gap
        else:
            if seg_start is not None:
                segments.append((seg_start, i - 1))
            seg_start, i = None, j
    if seg_start is not None:
        segments.append((seg_start, n - 1))
    return max(segments, key=lambda s: s[1]-s[0]) if segments else None
```

Output: `plot_bidask.py` on PERMNO 10145

Automatically starts 1992-12-28 — right after the 40-year gap — with no manual tuning required.

plot_returns.py: Core Logic

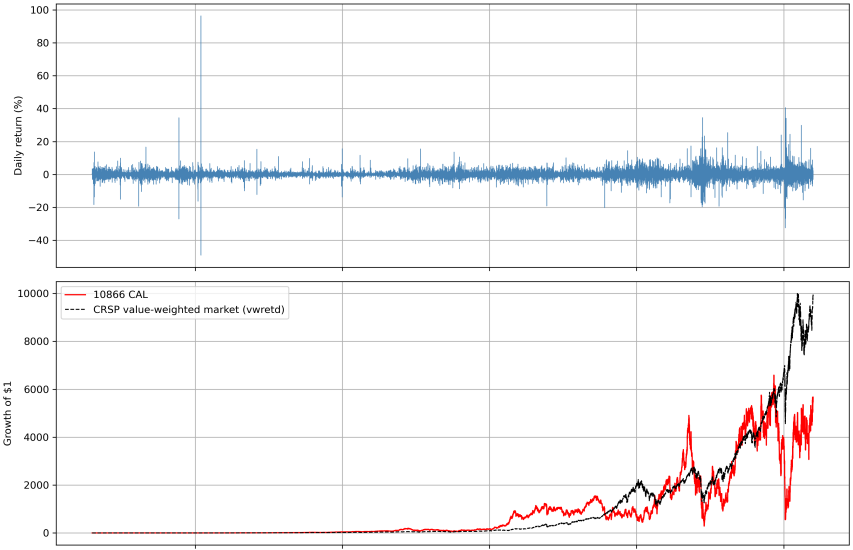
```
numeric_ret = pd.to_numeric(stock.RET, errors='coerce')  # letters -> NaN
stock['ret'] = numeric_ret.fillna(0.0)

cum_stock = (1 + stock.ret).cumprod()
cum_mkt   = (1 + stock.vwretd.fillna(0.0)).cumprod()

ax2.plot(xdata, cum_stock, 'r-', label=f'{permno} {ticker}')
ax2.plot(xdata, cum_mkt, 'k--', label='CRSP value-weighted market (vwretd)')
```

Days with a non-numeric `RET` code become 0% for the cumulative growth chart; the script prints how many such days it found.

Output: `plot_returns.py` on PERMNO 10866

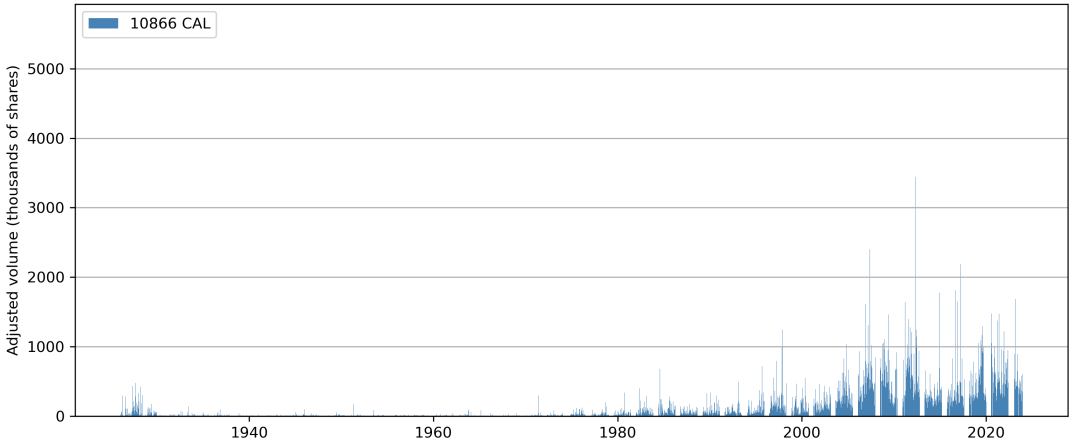


plot_volume.py: Core Logic

```
# VOL is raw (pre-split) shares traded; CFACSHR restates it in
# today's share-count terms so a split doesn't look like a volume spike
stock['adj_vol'] = stock.VOL * stock.CFACSHR

ax.bar(xdata, stock.adj_vol / 1000, width=1.5, color='steelblue')
plt.ylabel('Adjusted volume (thousands of shares)')
```

Output: `plot_volume.py` on PERMNO 10866



Outline

1. Introduction to CRSP
2. Our Local Dataset
3. Data Dictionary: Header & Name Identification
4. Data Dictionary: Distributions, Shares & Delisting
5. Data Dictionary: NASDAQ, Price/Volume/Return & Benchmarks
6. Coding Schemes
7. Key Conventions & Calculations
8. Data Wrangling in Python
9. Python Programs for Plotting
10. Wrap-Up

Try It Yourself

1. Open `listofstocks.txt` and pick any PERMNO.
2. Run all four scripts against it:

```
python plot_price.py   <PERMNO>
python plot_volume.py  <PERMNO>
python plot_returns.py <PERMNO>
python plot_bidask.py  <PERMNO>
```

3. Compare: does the price chart show visible split artifacts if you forget to divide by `CFACPR`?
4. Check whether your PERMNO has a large BID/ASK coverage gap like 10145's.
5. Pick a delisted security (`DLSTCD` $\neq 100$ on its last row) and look up its `DLSTCD` in the Coding Schemes section.

Where to Go Next

- **Full field reference:** *CRSP Data Description Guide for CRSPAccess (FIZ) US Stock & Index Databases* (July 2025), in this folder — Chapter 2 for every field's full definition, Chapter 5 for every coding scheme in full, Chapter 6 for the file-structure tables this deck's organization is based on.
- **Calculations:** Chapter 3 of the same guide documents CRSP's index methodologies in full, including exactly how `vwretd`/`ewretd` are built.
- **WRDS:** if you need data beyond this local extract (other date ranges, other CRSP products), Wharton Research Data Services hosts the same database with SAS/Python/R query tools.

Questions?

`C:\CRSP_daily` — `USStocks/<PERMNO>.csv` — `plot_*.py`