Chapter 2: Market and Fundamental Data – Sources and Techniques

1. Why Data is the Foundation of Algorithmic Trading

Before you can build any machine learning (ML) trading system, you need data—lots of it, and of good quality. Data is the raw material for every trading idea, every backtest, and every model. Without the right data, even the most sophisticated algorithms will fail.

Think of data as fuel:
Just as a car can't run on empty, a trading model can't work without data. The better your fuel (data), the smoother and faster your journey (model performance).

Types of questions data helps answer:

- What happened in the market yesterday?
- How did a company's profits change last quarter?
- Are investors feeling optimistic or fearful right now?

Key Point:
The more accurate, relevant, and timely your data, the more likely your trading ideas will work in the real world.

2. Types of Data Used in Trading

A. Market Data

This is the most basic and essential data for any trader. It tells you what's happening in the financial markets right now and what has happened in the past.

Market data includes:

- **Price data:** The price at which a stock, bond, or other asset is traded. This is usually given as open, high, low, and close prices for each day.
- **Volume data:** How many shares or contracts were traded.
- **Intraday data:** Prices and volumes at intervals shorter than a day (like every minute or second). This is crucial for high-frequency or intraday trading strategies.

Example:
If you look up Apple (AAPL) for July 1, 2024, you might see:

- Open: $180
- High: $182
- Low: $179
- Close: $181

- Volume: 20 million shares

B. Fundamental Data

This data describes the financial health and performance of companies. It's what investors use to judge whether a company is strong or weak, undervalued or overvalued.

Fundamental data comes from:

- **Financial statements:**
  - *Income statement* (shows revenue, expenses, and profit)
  - *Balance sheet* (shows assets, liabilities, and equity)
  - *Cash flow statement* (shows how cash moves in and out)
- **Key ratios and metrics:**
  - Earnings per share (EPS)
  - Price-to-earnings (P/E) ratio
  - Book value per share
  - Dividend yield

Example:
If a company earned $100 million last year and has 50 million shares, its EPS is $2.

C. Alternative Data

This is any non-traditional data that might give you an edge. It's not found in standard price or financial statement feeds.

Examples of alternative data:

- News headlines and articles
- Social media posts (Twitter, Reddit)
- Satellite images (like counting cars in a retailer's parking lot)
- Web search trends
- Credit card transaction data
- Weather data

Why use it?
Alternative data can reveal trends and signals before they show up in prices or official reports. For example, a spike in positive tweets about a company might predict a price jump.

3. Where to Find Financial Data

A. Free Data Sources

- **Yahoo Finance:**
  Offers free historical stock prices, basic financials, and some fundamental data.
  You can download data manually or use Python libraries like yfinance.

- **Alpha Vantage:**
  Provides free APIs for stock, forex, and cryptocurrency prices, plus some technical indicators.
- **Quandl:**
  Offers free and paid datasets, including prices, economic indicators, and more.
- **FRED (Federal Reserve Economic Data):**
  The go-to source for U.S. macroeconomic data, like unemployment rates, GDP, and interest rates.

B. Paid Data Sources

- **Bloomberg, Reuters, FactSet:**
  These are professional, institutional-grade data providers. They offer real-time prices, deep historical data, company fundamentals, news, and analytics. They're expensive but very comprehensive.
- **Specialty Vendors:**
  Some companies focus on alternative data, like satellite imagery, credit card data, or social media sentiment.

C. Accessing Data with Python

Most modern data providers offer APIs, which let you download data directly into your Python code. This is essential for automating your research and backtesting.

Example:
To get Apple's daily prices for the last year using yfinance:

python)

```
import yfinance as yf

data = yf.download('AAPL', start='2023-01-01', end='2024-01-01')

print(data.head())
```

This gives you a DataFrame with columns for Open, High, Low, Close, Volume, and Adjusted Close.

4. Cleaning and Preparing Data

Raw data is rarely ready to use. It might have missing values, errors, or inconsistencies. Cleaning and preparing your data—called **data preprocessing**—is critical before you can build any models.

A. Handling Missing Data

Why does data go missing?

- Markets are closed on holidays or weekends.
- Technical glitches or incomplete data feeds.
- Some companies don't report all financial metrics.

How to handle missing data:

- **Forward fill:** Fill missing values with the last known value (good for prices).
- **Interpolation:** Estimate missing values based on nearby data points.
- **Drop rows:** If only a few rows are missing, you might just remove them.

Example:
If there's no price for a stock on a holiday, you can use the last available price for that day.

B. Adjusting for Corporate Actions

Stock splits:
If a company splits its shares (e.g., 2-for-1), the price drops in half, but the value doesn't change. If you don't adjust for this, your price charts will have sudden, artificial drops.

Dividends:
When a company pays cash to shareholders, the stock price drops by the dividend amount. Adjusted prices account for this, so your analysis isn't thrown off.

Solution:
Use **adjusted close prices**—these automatically account for splits and dividends.

C. Data Alignment

If you combine multiple data sources (like prices and earnings), make sure the dates line up. You don't want to use future information by accident (this is called look-ahead bias).

D. Data Types and Consistency

Ensure all your data is in the right format (dates as dates, numbers as numbers). Consistent data types prevent bugs and errors in your analysis.

5. Exploring and Understanding Your Data

Before you build any models, you should **explore** your data to understand its structure, spot problems, and get a feel for what's going on.

A. Visualizing Data

- **Line plots:** Show how prices or indicators change over time.
  Example: Plotting Apple's closing price for the last year reveals trends and volatility.

- **Histograms:** Show the distribution of returns (how often do big moves happen?).
- **Scatter plots:** Compare two features, like volume versus price change, to see if there's a relationship.

B. Summary Statistics

- **Mean:** The average value.
- **Median:** The middle value.
- **Standard deviation:** How much values vary (measure of volatility).
- **Min/Max:** The highest and lowest values.

Why do this?
It helps you spot outliers, errors, or unusual behavior in your data. For example, if you see a stock price jump from $10 to $1000 in one day, it might be a data error.

C. Checking for Stationarity

Many ML models work best if the data's statistical properties (like mean and variance) don't change over time. This is called **stationarity**. If your data isn't stationary, you might need to transform it (for example, by using returns instead of prices).

6. Feature Engineering: Turning Raw Data into Model Inputs

Feature engineering is the process of creating new, more informative variables (features) from your raw data. Good features help your model learn useful patterns.

A. Technical Indicators

These are formulas applied to price and volume data to highlight trends, momentum, or volatility.

Common technical indicators:

- **Moving averages:** Smooth out price data to spot trends.
  Example: A 20-day moving average is the average closing price over the last 20 days.
- **Relative Strength Index (RSI):** Measures if a stock is overbought (high RSI) or oversold (low RSI).
- **MACD (Moving Average Convergence Divergence):** Shows changes in momentum.

How to use:
If today's price is above the 20-day moving average, it might signal an uptrend. If RSI is above 70, the stock might be overbought.

B. Fundamental Ratios

These are calculations based on financial statements.

Examples:

- **P/E ratio:** Price divided by earnings per share. High P/E might mean a stock is expensive.
- **Debt/equity ratio:** How much debt a company has compared to its equity.

How to use:
A low P/E ratio might indicate a bargain, while a high debt/equity ratio could signal risk.

C. Alternative Data Features

- **Sentiment score:** Use natural language processing (NLP) to turn news headlines or tweets into a number from -1 (very negative) to +1 (very positive).
- **Event flags:** Mark days with big news events, like earnings announcements or product launches.

D. Lagged Features

To avoid look-ahead bias, you often use lagged features—values from previous days or weeks.

Example:
Use yesterday's closing price or last week's average volume as features to predict tomorrow's price.

7. Common Data Challenges and How to Avoid Them

A. Survivorship Bias

If you only use data from companies that exist today, you ignore those that went bankrupt or got delisted. This makes your backtests look better than reality.

How to avoid:
Use historical lists of all stocks, including those that disappeared.

B. Look-Ahead Bias

This happens if you accidentally use data that wasn't available at the time you're making a prediction.

Example:
Using next quarter's earnings to predict today's price.

How to avoid:
Always use only the data that would have been available at each point in time.

C. Data Snooping (Overfitting)

If you try lots of ideas on the same data, some will look good just by chance.

How to avoid:
Keep a separate “holdout” dataset that you only use for final testing.

D. Incomplete or Dirty Data

Missing or incorrect data can lead to wrong conclusions.

How to avoid:
Always check and clean your data before modeling.

8. End-to-End Example: Preparing Data for a Trading Model

Let’s walk through a practical workflow:

Step 1: Choose a stock
For example, Apple (AAPL).

Step 2: Download daily price data
Use yfinance to get 5 years of closing prices, volumes, and adjusted close prices.

Step 3: Clean the data

- Handle missing values with forward fill.
- Use adjusted close prices to account for splits and dividends.

Step 4: Create features

- Calculate a 10-day moving average.
- Compute a 14-day RSI.
- Add yesterday’s trading volume.

Step 5: Add fundamental data

- Download quarterly earnings per share (EPS) and P/E ratio from Yahoo Finance or another source.
- Align the data by date (for example, use the latest available EPS for each date).

Step 6: Add alternative data

- Download news headlines for Apple.
- Use a simple sentiment analysis tool to score each headline.
- Aggregate daily sentiment scores.

Step 7: Align all features
Make sure every row in your dataset has the correct price, technical indicators, fundamental ratios, and sentiment score for that date.

Step 8: Explore and visualize

- Plot price and moving average to spot trends.
- Plot RSI to see overbought/oversold conditions.
- Check summary statistics for all features.

Step 9: Save your clean dataset

Now you have a feature-rich, cleaned dataset ready for modeling in the next chapters.

9. Key Takeaways

- **Data is the foundation:** The quality and preparation of your data is more important than the complexity of your model.
- **Use multiple data types:** Combine market, fundamental, and alternative data for richer features.
- **Clean and align everything:** Handle missing values, adjust for splits/dividends, and make sure all features line up by date.
- **Engineer informative features:** Technical indicators, fundamental ratios, and sentiment scores can all help your model.
- **Avoid common pitfalls:** Survivorship bias, look-ahead bias, and data snooping can make backtests look better than real results.
- **Always explore your data:** Visualization and summary statistics help you spot problems before modeling.

Home work 1

- **Practice downloading data:** Use Python libraries like yfinance, pandas_datareader, or APIs from Alpha Vantage and Quandl.
- **Try cleaning real data:** Handle missing values, adjust for splits, and create technical indicators.
- **Calculate and plot features:** Moving averages, RSI, and sentiment scores.
- **Experiment with combining data:** Merge price, fundamental, and alternative data for a single stock.
- **Visualize everything:** Plot prices, indicators, and distributions to get a feel for your data.