📚Quantitative Trading & Portfolio Analytics — Master Notes

This document unifies **all your notebooks** (01–10) into a single, comprehensive study guide. It covers everything from Pandas basics to portfolio optimization, CAPM, pipelines, leverage, hedging, sentiment, and futures.

✂️Pandas Fundamentals & Data Access

Key Concepts

- **Series and DataFrames** — core data structures in Pandas.
- **Indexing:** `.loc[]` for label-based, `.iloc[]` for position-based.
- **Selection & Filtering:** Boolean masks, `.query()` for readability.
- **Missing Data:**
- `dropna()` — remove rows/columns with NaN.
- `fillna()` — replace NaN with mean, median, 0, forward/backward fill.
- **GroupBy:** aggregate by category (e.g. sector, date).
- **Resample:** change frequency of time series (daily → monthly).

Example

```
md.dropna(thresh=2)  # keep rows with >=2 non-NaN values
md.fillna({'A': md['A'].mean(), 'B': 0})
```

✒️Pandas Time Series Mastery

Key Operations

- **DatetimeIndex:** convert with `pd.to_datetime()`, set as index.
- **Shifting & Lagging:** `.shift()` to align features (avoid look-ahead bias).
- **Rolling Windows:** `.rolling(window=20).mean()` for SMA, `.std()` for volatility.
- **Resampling:** `.resample('M').sum()` to aggregate monthly.

SMA vs EMA

- **SMA:** Simple average over N periods.
- **EMA:** Exponentially weighted average → more weight to recent data.
- EMA reacts faster and avoids initial NaN block.

🛏️Visualization & Plotting

Plot Types

- **Area:** trend of cumulative values (use `stacked=False` if data has + and -).
- **Bar / Stacked Bar:** compare categories.
- **Histogram:** visualize distribution.
- **Line (time index):** trend analysis.
- **Scatter:** relationship between two variables, use `c=` and `s=` for color & size.
- **Boxplot:** median, quartiles, outliers.
- **KDE:** smooth probability density function.
- **Hexbin:** bivariate density for large datasets.

Tip

Always set titles via `ax.set_title()` and call `plt.tight_layout()` to avoid overlap.

🪁Time Series Modeling (ETS, EMA, ARIMA)

ETS Decomposition

Splits series into: - **Trend:** long-term movement - **Seasonality:** repeating patterns - **Residuals:** unexplained noise

Stationarity & Differencing

- **ADF Test:** p-value < 0.05 ⇒ stationary.
- If not stationary → difference once (`df.diff()`) or seasonally.

ARIMA Modeling

- Choose `(p,d,q)` based on ACF/PACF plots.
- Fit using `statsmodels` ARIMA.
- Check residuals → should look like white noise.

🌡️ Guided Stock Market Analysis

Steps

1. Load OHLCV data.
2. Plot `Open`, `Volume`, `TotalTraded`.
3. Compute **MA(50)** and **MA(200)**, look for crossovers (Golden Cross / Death Cross).
4. Calculate **daily returns** and plot histograms/KDE.
5. Compute **rolling volatility** to measure risk.
6. Plot **scatter matrix** and compute correlations → key for diversification.
7. Calculate **Cumulative Daily Returns (CDR)** to see long-term growth of $1 invested.

🎒Stock Market Resumen (Quick Reference)

- **Volume spikes** = potential news/events.
- **Golden Cross:** bullish trend confirmation.
- **Histogram width:** indicates volatility.
- **Correlation matrix:** find low correlation pairs for diversification.

🎓Quantopian Mini-API & Backtesting

Context Object

Simulates Quantopian environment: - **Positions & cash** - **Record()** to log metrics - **Order_target()** to simulate trades

Pairs Trading Example

- Compute spread between two stocks.
- Calculate z-score: `(spread - mean)/std`.
- Go long/short based on thresholds (e.g. ±1σ).

Bollinger Bands

- SMA ± k×σ → detect overbought/oversold.
- Buy near lower band, sell near upper band.

🎩Portfolio Analytics & CAPM

Portfolio Math

- **Return:** `w @ mu`
- **Volatility:** `sqrt(w @ Σ @ w)`
- **Sharpe Ratio:** `(r - rf) / vol`

Monte Carlo Simulation

- Generate random weights, plot risk-return cloud.
- Identify efficient frontier (upper envelope).
- Find max Sharpe portfolio.

CAPM

$$E[R] = R_f + \beta(R_m - R_f)$$

- β > 1 = more sensitive to market moves. - Use linear regression of portfolio vs market.

🎽 Advanced: Rebalancing, Leverage & Hedging

Rebalancing

- Align portfolio weights periodically (weekly, monthly).
- Trade-off: more frequent = higher costs but faster reaction.

Leverage

- Scales weights: `w_lever = L * w_base`.
- Controls **gross exposure** (sum of abs weights) and **net exposure** (sum of weights).
- Leverage amplifies returns *and* volatility.

Hedging (CAPM Beta Neutralization)

- Estimate portfolio β vs market.
- Add hedge position: `-β * weight_market`.
- Goal: keep exposure market-neutral, focus on alpha.

🕐 Pipelines, Sentiment & Futures (Advanced)

Pipelines

- Combine **factors**, **filters**, **classifiers** to build universe.
- Rank by signal, select top/bottom N.

Sentiment Signals

- Trigger trades on high-impact events with strong polarity.
- Use cooldown windows to avoid overtrading.

Futures Rolling

- Build continuous series using volume-based roll.
- Adjust for gaps (back-adjusted series).
- Watch out for roll yield and slippage.

💹 Final Checklist for Quant Strategies

✅Clean and align data (no NaNs, no look-ahead bias) ✅Compute features with proper shifting ✅Validate stationarity before ARIMA/mean reversion ✅Simulate with realistic costs and slippage ✅Analyze Sharpe, CAGR, Max Drawdown, Turnover ✅Stress-test with different rebalance frequencies & leverage ✅Visualize results: equity curve, drawdown curve, efficient frontier

> **Key Insight:** Robust strategies remain profitable under small parameter changes and across market regimes.