AMP Volatility Managed Portfolios

Moana Valdenaire, Wiktor Kotwicki, Nicolas Gamboa Alvarez

30th of November 2024

Where are we?

- Clearly defined the asset class we want to explore
- Able to provide the list of commodities we are going to work with
- Created a code library to support our research and findings
- Scheduled regular meetings with the team members

Article

We have focused for this session on the following article:

- Moreira, A., & Muir, T. (2017). Volatility-managed portfolios. *The Journal of Finance, 72*(2), 651-688. https://doi.org/10.1111/jofi.12423

Key findings

Key Findings

Description of Moreira and Muir Paper:

- The first paper to popularize the volatility-timed approach for improving risk-adjusted returns.
- Investors reduce risk exposure during periods of high realized volatility and increase it when volatility is low.
- Contrasts traditional risk-return trade-off assumptions.

Key Findings from the Paper:

- **Improvement in Sharpe Ratios:** Higher risk-adjusted returns across portfolios.
- **Resilience in Recession Periods:** Reduced drawdown due to lower exposure during volatile times.
- **Utility Improvements:** Better outcomes for mean-variance investors and long-term wealth accumulation.
- **Expansion of Mean-Variance Frontier:** Broader opportunities for portfolio optimization.

Methodology and Data

Variance and Volatility Measures:

- Inverse of the previous month's realized variance as a primary measure.
- Alternatives explored:
  - Previous month's realized variance or volatility.
  - Expected variance and strategies without leverage or with 50% leverage.

Volatility-Managed Portfolios Factor Equation:

$$f^{\sigma}_{t+1} = \frac{c}{\hat{\sigma}^2_t(f)} f_{t+1},$$

Data and Assets:

- Monthly data reduces rebalancing frequency.
- Standard Fama-French factors (MKT, SMB, HML, ...) from US datasets.
- Robustness tested with credit risk factors, corporate bonds, and currencies.

Transaction Costs and Other Challenges

Transaction costs:

- High turnover erodes Sharpe ratio due to transaction costs, especially for illiquid factors (Barroso & Detzel, 2021).
- Frequent rebalancing and high costs challenge practical implementation.

Broader Implications:

- Best performance during high-sentiment periods, but market conditions limit robustness.

Practical Challenges:

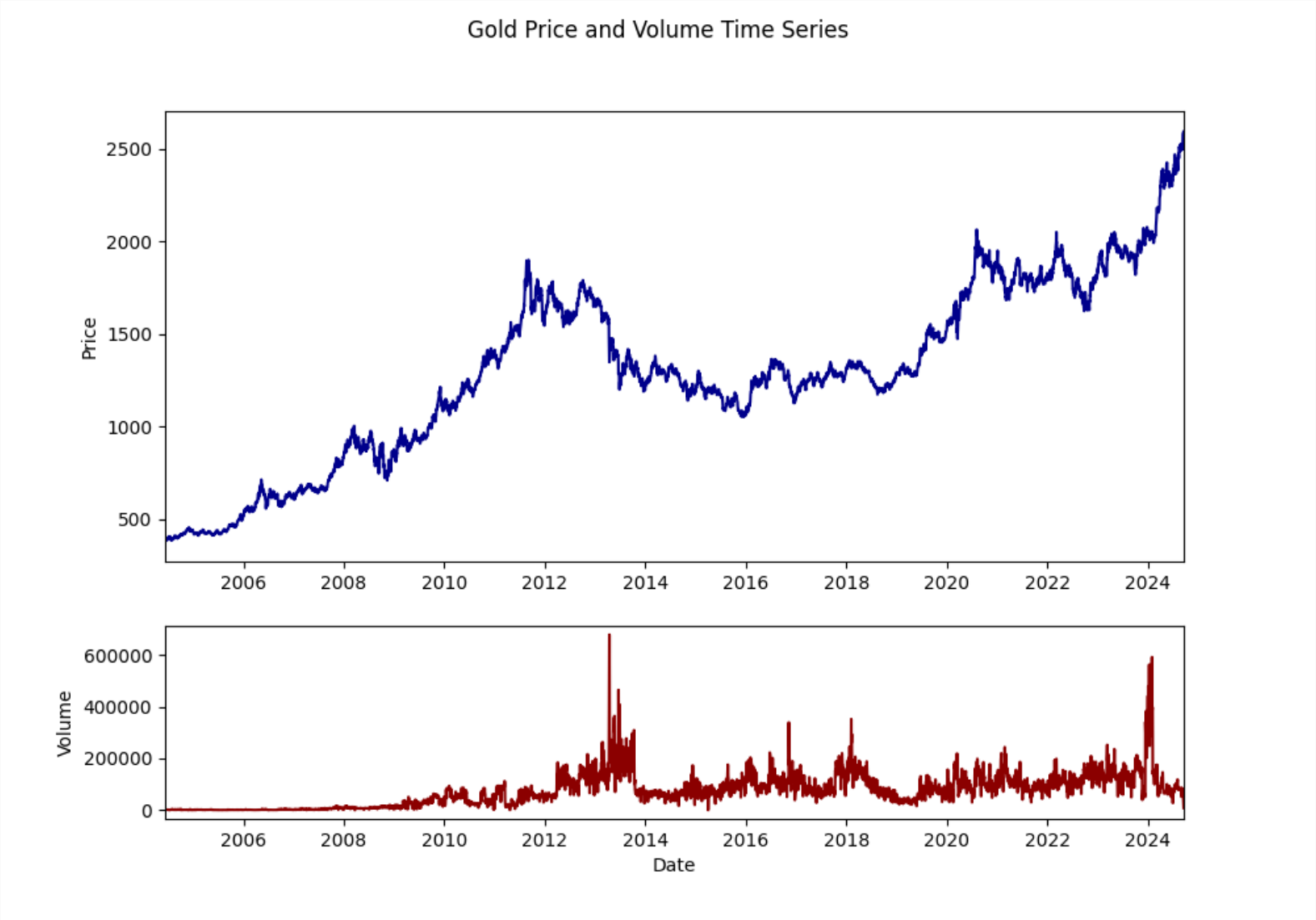
- Look-ahead bias and instability lead to underperformance (Cederburg, O'Doherty, & Jiang, 2020).
- High transaction costs and sentiment dependence limit long-term success (Barroso & Detzel, 2021).

Discussion and Future Research

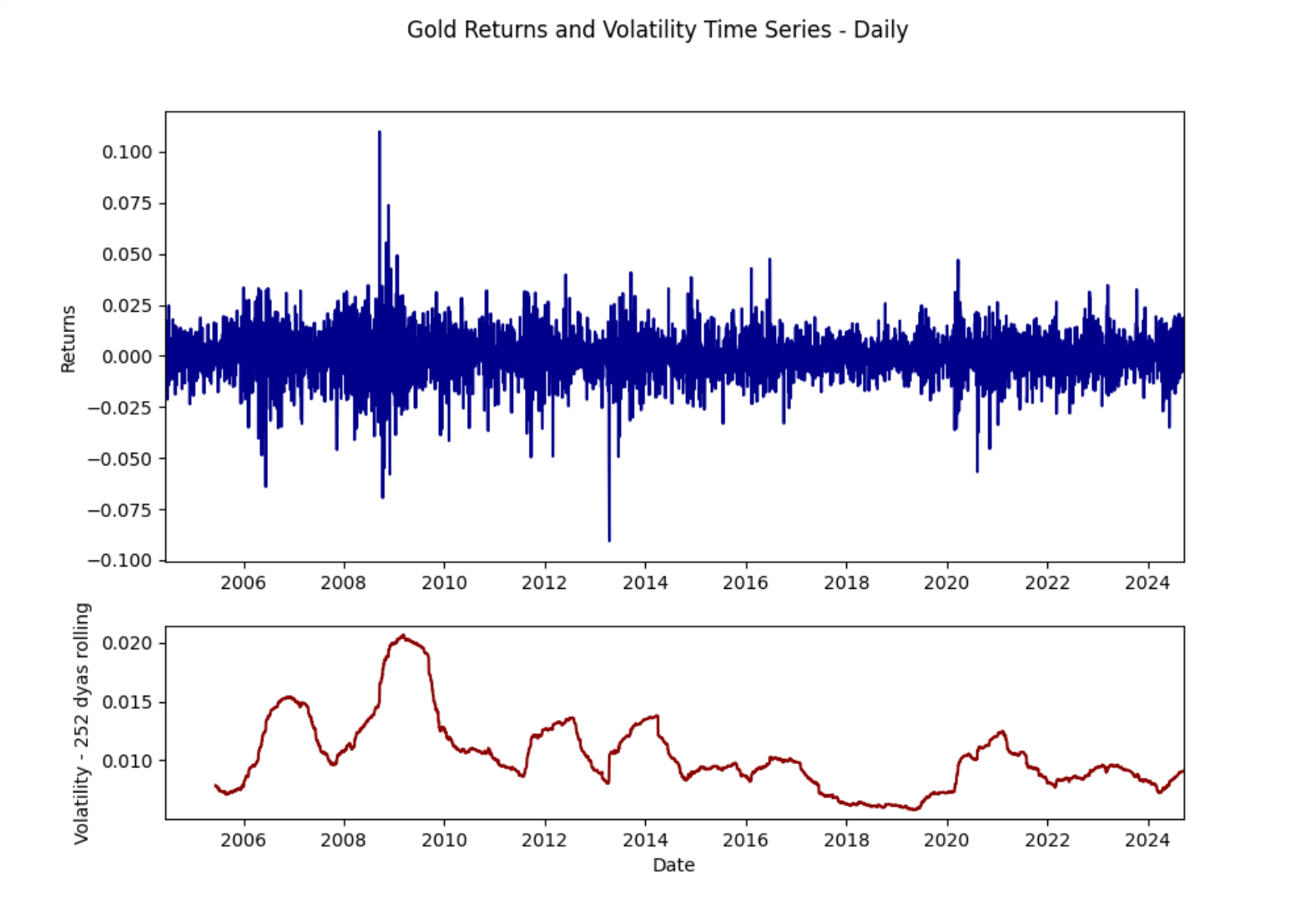
Big Picture Question/Idea:

- How can we better account for transaction and turnover costs?
- Which commodities offer the best testing ground for these strategies?
- Can we define recession indicators tailored to chosen commodities?
- What data frequency provides the optimal balance between signal reliability and costs?

Visualization I: Gold Prices and Volumes (Daily)



Visualization II: Gold Returns Volatility (Daily)



What is next?

To do next:

- Going through more research
- Schedule a session in the Bloomberg room
- Retrieve and explore historical data from commodities
- Building the necessary code to implement portfolio construction dynamics
- Continue meeting regularly with the team

What we are building - A code framework

volman_module.py > ...

```python
# ------------------------------------------------------------------------------------------------------
# This file contains a series of functions used for processing, analyzing, and visualizing data
# in the context of volatility managed portfolios. It also contains functions for portfolio
# construction, optimization, and backtesting - including adjustments for volatility management.
# This is intended as a module to be imported into other scripts or notebooks.
# This module is part of the Volatility Managed Portfolio project for EDHEC Business School.
# Nicolas Gamboa Alvarez, Wiktor Kotwicki, Moana Valdelaire, 2024 - 2025
# ------------------------------------------------------------------------------------------------------

# Importing necessary libraries ------------------------------------------------------------------------
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Functions --------------------------------------------------------------------------------------------

# Function that takes a dataframe, optionally takes a column name. Then it should return the
# dataframe as a time series with the datetime column as index.
def to_time_series(df, col_name='Date', format='%Y-%m-%d'):
    """
    Converts a specified column in a DataFrame to datetime format and sets it as the index.
    Parameters:
    df (pandas.DataFrame): The DataFrame to convert.
    col_name (str): The name of the column to convert to datetime format. Default is 'Date'.
    format (str): The datetime format to use for conversion. Default is '%Y-%m-%d'.
    Returns:
    pandas.DataFrame: The DataFrame with the specified column converted to datetime format and se
    Raises:
    ValueError: If the specified column does not exist in the DataFrame or if the conversion to d
    """
```