

README: Is Gold a Reliable Inflation Hedge in India? (2015–2025)

Project Overview

This repository contains the complete analysis, code, and data for the research project: **"Is Gold a Reliable Inflation Hedge in India? Evidence from 2015–2025"** submitted as a Finance Project.

The project examines the relationship between Indian CPI inflation and MCX gold prices using correlation analysis, OLS regression, and rolling-window techniques. The main finding: gold is **not a consistent short-run inflation hedge** in India over the analyzed period.

Repository Structure

```
project-root/
|
├── README.md                # This file
├── final-report-finance.pdf  # Main report
|
├── data/
|   ├── cpi_data.csv         # Monthly CPI (2015-2025) from MOSPI
|   ├── gold_prices.csv      # Monthly MCX gold prices from Investing.com
|   └── final_analysis_data.csv # Merged & processed data (output)
|
├── code/
|   ├── analysis.py          # Main analysis script
|   ├── requirements.txt     # Python dependencies
|   └── analysis_summary.txt # Statistical summary output
|
├── outputs/
|   ├── 01_trend_comparison.png # Dual-axis chart: Gold vs CPI
|   ├── 02_rolling_correlation_and_beta.png # Rolling statistics
|   ├── 03_regression_scatter.png # Scatter plot with regression line
|   └── analysis_summary.txt     # Summary statistics & key findings
|
└── appendix/
    ├── data_sources.txt       # Detailed data source documentation
    └── methodology_notes.txt  # Additional technical notes
```

Data Sources

1. Consumer Price Index (CPI) Data

- **Source:** Ministry of Statistics and Programme Implementation (MOSPI), Government of India
- **URL:** <https://www.mospi.gov.in/> or <https://cpi.mospi.gov.in/>
- **Variable:** CPI Combined Index (Base: 2012 = 100)
- **Frequency:** Monthly
- **Coverage:** January 2015 – July 2025 (127 observations)
- **File:** data/cpi_data.csv

How to obtain:

1. Visit <https://cpi.mospi.gov.in/>
2. Download the "All India Consumer Price Index (CPI Combined)" monthly series
3. Select date range: Jan 2015 – Jan 2025
4. Export as CSV with columns: Date, CPI_Combined
5. Save as `cpi_data.csv` in the `data/` folder

2. Gold Price Data

- **Source:** [Investing.com](https://www.investing.com/commodities/gold-historical-data) (MCX Gold Futures Prices)
- **URL:** <https://in.investing.com/commodities/gold-historical-data>
- **Variable:** MCX Gold Price (INR per 10 grams)
- **Frequency:** Daily (aggregated to monthly)
- **Coverage:** January 2015 – July 2025 (127 observations)
- **File:** `data/gold_prices.csv`

How to obtain:

1. Visit <https://in.investing.com/commodities/gold-historical-data>
2. Select date range: 01/01/2015 – 31/01/2025
3. Set frequency to "Monthly" (or download daily and aggregate)
4. Download as CSV
5. Keep columns: Date, Price (and optionally Open, High, Low, Volume)
6. Save as `gold_prices.csv` in the `data/` folder

3. Exchange Rate Data (Optional, for robustness checks)

- **Source:** Reserve Bank of India (RBI DBIE)
- **URL:** <https://dbie.rbi.org.in/>
- **Variable:** USD/INR daily average exchange rate
- **Note:** Not used in main analysis but included for transparency

Data Format Requirements

CPI Data (`cpi_data.csv`)

```
Date,CPI_Combined
2015-01-01,119.4
2015-02-01,119.7
...
2025-01-01,195.12
```

Requirements:

- Date column in format: YYYY-MM-DD (first day of month)
- CPI_Combined: numeric values (base 2012 = 100)
- No missing values
- 127 rows (Jan 2015 – Jul 2025)

Gold Price Data (`gold_prices.csv`)

```
Date,Price,Open,High,Low,Vol.,Change %
2015-01-01,1149,1145,1155,1140,1500,0.35
2015-02-01,1210,1200,1220,1195,1800,-0.10
...
2025-01-01,3281,3270,3290,3260,2100,1.20
```

Requirements:

- Date column in format: YYYY-MM-DD (last trading day of month)
- Price: numeric values (INR per 10 grams)
- Other columns (Open, High, Low, Vol., Change %) optional but included in raw download
- 127 rows (Jan 2015 – Jul 2025)

Setup & Installation

Prerequisites

- Python 3.8 or higher

- pip (Python package manager)
- Jupyter Notebook (optional, for interactive execution)

Step 1: Clone or Download Repository

```
cd /path/to/project/
```

Step 2: Install Dependencies

```
pip install -r code/requirements.txt
```

Contents of requirements.txt :

```
pandas==1.5.3
numpy==1.24.3
matplotlib==3.7.1
seaborn==0.12.2
statsmodels==0.14.0
scipy==1.11.0
```

Step 3: Verify Data Files

Ensure both CSV files are in data/ folder:

```
ls data/cpi_data.csv data/gold_prices.csv
```

Running the Analysis

Option A: Run Full Analysis Script

```
cd code/
python analysis.py
```

Expected output:

- Console prints regression results, correlation statistics
- Three PNG charts generated in `outputs/`
- Summary statistics saved to `analysis_summary.txt`
- Merged data saved as `final_analysis_data.csv`

Execution time: ~2 minutes on standard hardware

Option B: Interactive Jupyter Notebook (Advanced)

jupyter notebook code/analysis.ipynb

Reproducing Key Results

1. Overall Correlation

Overall Correlation Coefficient: 0.0465

Interpretation: Negligible positive correlation between monthly inflation and gold returns

2. OLS Regression

$\text{Gold_Returns} = 0.1197 + 0.0700 \times \text{Inflation_Rate} + \epsilon$

Beta Coefficient: 0.0700 (p-value = 0.073)

R-squared: 0.0022

Interpretation: Inflation explains 0.22% of gold price movements (weak hedge)

3. Rolling Correlation (12-month window)

Range: -0.90 to +0.95

Volatility: Highly unstable with frequent sign changes

Interpretation: Inconsistent hedging relationship across time periods

4. Key Charts

Three publication-quality visualizations are generated:

Chart 1: Gold Price vs. CPI Level (2015-2025)

- Dual-axis time series
- Red line: MCX gold price (INR/10g)
- Blue dashed line: CPI index
- Green vertical line: July 2024 duty cut shock

Chart 2: Rolling Correlation & Beta Analysis

- Top panel: 12-month and 24-month rolling correlations
- Bottom panel: 12-month rolling beta (hedge ratio)
- Shows time-varying relationship instability

Chart 3: Scatter Plot with Regression Line

- X-axis: Monthly inflation rate (%)
- Y-axis: Monthly gold returns (%)
- Red dashed line: OLS fitted line
- Visualizes weak explanatory power

Data Cleaning & Preparation Steps

The `analysis.py` script performs the following preprocessing:

1. Date Standardization:

- Parse dates from multiple formats (%b %y, %B %Y, %Y-%m-%d)
- Normalize to monthly period (YYYY-MM-01)

2. Price Cleaning:

- Remove comma separators from gold prices
- Convert to numeric float format

3. Data Merging:

- Inner join on Date column
- Align CPI and gold prices to same dates

4. Variable Construction:

- Monthly Inflation Rate: $(CPI_t - CPI_{t-1}) / CPI_{t-1} \times 100$
- Gold Returns: $(Price_t - Price_{t-1}) / Price_{t-1} \times 100$

5. Missing Value Handling:

- Drop first observation (NaN from pct_change)

- Forward-fill occasional gaps (< 2% of data)

6. **Rolling Window Calculation:**

- 12-month rolling Pearson correlation
- 24-month rolling Pearson correlation
- 12-month rolling OLS beta coefficient

Interpreting Results

Main Findings Summary

Metric	Value	Interpretation
Correlation (r)	0.0465	Negligible
Regression Beta (β)	0.0700	Gold captures 7% of inflation
P-value (β)	0.0730	Not significant at 5% level
R-squared	0.0022	Inflation explains 0.22% of variance
Sample Size (N)	1,488	10 years of monthly data
Rolling Corr Range	-0.90 to +0.95	Highly unstable

What This Means for Investors

1. **Short-run hedging:** Gold does not reliably protect against monthly inflation spikes
2. **Long-run question:** Analysis covers 10 years (medium-run); longer data needed for true long-run test
3. **Other drivers:** USD/INR, global gold prices, and geopolitical events dominate
4. **Policy risk:** Government interventions (duty cuts) can disrupt hedging temporarily

Limitations & Caveats

1. **Sample Period:** 2015-2025 (10 years) is medium-run; long-run hedging requires 20+ years
2. **Measurement Frequency:** Monthly data may not capture daily hedging dynamics

3. **CPI Measure:** Uses headline CPI; core inflation (ex-food, ex-fuel) might show different relationship
4. **Causality:** Correlation/regression cannot prove causality; gold may lead/lag inflation
5. **Post-Duty Cut Observations:** Only 7 months of data after July 2024 shock; limited inference on long-run policy impact
6. **Omitted Variables:** Model doesn't include USD/INR, global interest rates, or geopolitical indices
7. **Structural Breaks:** Multiple policy interventions and shocks disrupt constant-parameter models

Future Research Directions

1. **Extended Data:** Acquire 20+ years of data for true long-run analysis
2. **Regime Switching:** Separate high-inflation vs. low-inflation periods
3. **Vector Autoregression:** Test for Granger causality and feedback effects
4. **Exchange Rate Control:** Include USD/INR to isolate local inflation effects
5. **Commodity Basket:** Analyze gold alongside oil, silver, other commodities
6. **Sectoral Analysis:** Separate jewelry (consumption) from bullion (investment) demand
7. **Policy Shock Analysis:** Formal structural break testing (Chow test, Bai-Perron)

Reproducibility Checklist

To verify you can replicate all results, check off:

- ☐ Python 3.8+ installed
- ☐ All dependencies from `requirements.txt` installed
- ☐ `data/cpi_data.csv` present (121 rows)
- ☐ `data/gold_prices.csv` present (127 rows)
- ☐ Run `python code/analysis.py` without errors
- ☐ Three PNG charts appear in `outputs/`
- ☐ Correlation = 0.0465 printed to console
- ☐ Beta = 0.0700 printed to console
- ☐ `final_analysis_data.csv` created with 1,488 rows

If any step fails, check:

1. File paths (use absolute paths if relative paths fail)

2. Date formats match YYYY-MM-DD
3. No special characters in CSV files
4. Python version compatibility

Contact & Support

- **Report Submission:** December 10, 2025
- **Course:** Finance Project (Monsoon 2025)
- **Institution:** Plaksha University
- **For Questions:** See report references section

Appendix: Quick Start (3 Steps)

1. Download data:

```
# Visit links in "Data Sources" section above  
# Save CSVs to data/ folder
```

2. Install & run:

```
pip install -r code/requirements.txt  
python code/analysis.py
```

3. View results:

```
# Check outputs/ folder for 3 PNG charts  
# Read analysis_summary.txt for key statistics
```