Gold and cryptocurrencies exchange rates correlation

Python Project

Polytechnique Institute of Paris
Applied Mathematics and Statistics - Master degree
Python for Datascience - Anas Hattay
Fall Semester 2025 - M1 APPMS

Zhousiying Wu, Nicolas Valety
17.12.2025 - v0.1.0 - draft

Contents

1 Introduction

This chapter provides an overview of the project, including its background, motivation, and objectives. It should clearly state the problem being addressed and why it is relevant.

Key elements:
- ***Introduce the goal** – What do you want to achieve with the project?*
- ***Provide context** – Why is this project relevant? What problem does it solve?*
- ***Define the scope** – What are the boundaries and limitations of your project?*
- ***Outline the structure** – How is this report organized?*

Always place yourself in the point of view of the reader. For who is the report intended? What do they need to know to understand the project? Create and follow a red thread that guides the reader through the report.

1.1 Context

Gold has been used as currency since the 6th century BC and has played an important role throughout world history. Especially after the establishment of the gold standard, gold becomes the basis for maintaining the world monetary system. Later, although the gold standard system was gradually abolished in the 20th century due to the Great Depression, gold still occupies an important position in the modern economy. Conversely, cryptocurrencies are much more recent and have been launched in the 21th century on the Blockchain. Even if the gold standard system has been abolished, does any correlation between gold and cryptocurrencies could be found? An analysis on several period of time in order to find out one (or more) year when gold stop (or begin) to act as a standard for cryptocurrencies. Our research will aim in a procedure to find it out by defining objectives, enumerating methods needed and the toolkit compulsory to compute data.

1.2 Aims and Objectives

1.2.1 Aims: We want to find out a correlation between gold and any cryptocurrency exchange rates.

> At this point, there may not be any correlation between both exchanges rates as one is very stable and the other is very volatile.

1. Objective 1: Compare the exchange rates, find out if there is a comparable period of time when the cryptocurrency and gold are correlated.
2. Objective 2: Identify some ranges which contain (or do not) periods of correlation between gold and cryptocurrencies.

1.3 Methods

Data we will use is:
1. Date of observation
2. Name if currency
3. Currency/gold price at open market, close market and average day price

Additionally we want to find out other parameters (mean, variance, etc.) in order to compare the different exchanges using mathematical tools.

1.4 Toolkit

We will use some libraries for doing mathematical computations and to draw graphs (eg. Matplotlib).

1.4.1 Libraries used

> kagglehub and pandas are directly handled by the Dataset class

1. kagglehub (downloading latest datasets from Kaggle)
2. yfinance (downloading latest datasets from Yahoo! Finance)
3. pandas (to manipulate the dataframes)
4. matplotlib (to plot the charts and graphs)
5. seaborn (to show correlation matrices)

1.4.2 Datasets used

1. Crypto cryptocurrencies daily prices
2. Yahoo cryptos
3. Gold historical data daily updated
4. Yahoo! gold

1.4.2.1 Dataset modifications

1. Dates before 17/07/2010 must be droped as no cryptocurrency were available before
2. Columns keeped : Every columns from the gold dataset must be renamed with a lowercase name.
   - ticker: "XAU" value added to gold exchange rates
   - date: date format must be converted to yyyy-mm-dd for gold exchange rates dataset
   - open
   - high
   - low
   - close

1.4.3 Usefull tools

1. Git
   - git add newFile
   - git commit -a -m "comment"
   - git push -u origin main
   - git pull git@github.com:user/repository.git

2 Implementation

(to be removed) *This chapter details the development and execution of the project. It describes actual implementation from a top-down or bottom-up approach depending on the project.*

Key elements:

- ***Describe the implementation*** *– How was the project developed?*
- ***Explain technical decisions*** *– What tools, frameworks, or methods were used and why?*
- ***Highlight key components*** *– What are the most important parts of the implementation?*
- ***Address challenges*** *– What difficulties arose, and how were they solved?*

- Naive model: This is chosen as a baseline, the logic here is tomorrow's price = today's price. Because gold prices fluctuate very little in the short term,

I think this is a good benchmark.

- ARIMA: Because gold prices tend to revert to the mean in the long run, prices fluctuate over time, and past prices influence future prices. ARIMA can capture

the linear trend and autocorrelation of gold price changes.

- SARIMA: SARIMA and ARIMA operate on similar principles, but SARIMA incorporates seasonal patterns. Because gold prices are linked to seasonal demand

(e.g., as holiday gifts or wedding jewelry) and cultural patterns (e.g., the Indian wedding season and Chinese New Year), we believe economic cycles may exist. Additionally, the seasons in mining areas also affect production, thus influencing prices.

- Deep learning :

Evaluation: Naive model: ARIMA: SARIMA:

Then we add exgeous features to improve the predict, the exgeous feature we generated are

- day of week (0:monday - 6:sunday), because gold trading might have weekly pattern
- which week in the year(1-52), because gold jewelry trading might increase in certain months (e.g.chinese new year, indian wedding season)
- month of year(1-12), days, months, years
- month since start: As the data is started from 2022-05-16 to 2024-12-09, so this is how many months have passed since our data is started. Because it is obvious that gold price

have long-term upward trend. And models gradual inflation effects over time.

- if that day is a holiday / how many days to holiday: Since markets are closed for holidays and there is no trading data, we use the average price before and after the holidays to fill

the gap. Additionally, many investors adjust their positions before holidays, which also contributes to gold price fluctuations. The uncertainty during major holiday closures also affects gold prices.

2.1 Basic markup

Typst lets you create bold, italic, or monospaced text with ease. You can also sprinkle in equations like $e^{i\pi} + 1 = 0$ or even inline code like `fn main() { println!("Hello, World!") }`. And because life is better in color: pink, blue, yellow, orange, green, and more! Boldly colorize!

You can also write numbered or unnumbered lists:

- First item
- Second item
  1. First Subitem
  2. Second Subitem
- Third item

Need equations? Sure! They look great as blocks too:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \ldots = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1} \tag{1}$$

2.2 Images

As they say, a picture is worth a thousand words. Let's add one:

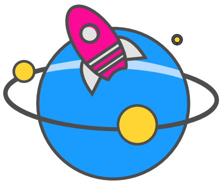

Figure 2 - Project logo

2.3 Tables

Tables are great for organizing data. From simple to complex, Typst handles them all:

Name	**Age**	**City**
Albert Einstein	25	Bern
Marie Curie	22	Paris
Isaac Newton	30	London

Table 1 - Simple table

[31:27]			[24:20]	[19:15]	[14:12]	[11:7]	[6:0]
funct5	aq	rl	rs2	rs1	funct3	rd	opcode
5			5	5	3	5	7

Table 2 - Complex table

2.4 Boxes

Highlight key points with these fun boxes (and more):

Infobox: For highlighting information.

Ideabox: Share a brilliant idea.

Warningbox: Proceed with caution!

Firebox: This is 🔥!

Rocketbox: Shoot for the moon!

Todobox: Just do it!

2.5 Citations, Acronyms and Glossary

Add citations with `@` like [1] (stored in `/tail/bibliography.bib`).

Acronym terms like Infotronics (IT) expand on first use and abbreviate after IT. Glossary items such as Rust Programming Language (Rust) can also be used to show their description as such: Rust is a modern systems programming language focused on safety, speed, and concurrency. It prevents common programming errors such as null pointer dereferencing and data races at compile time, making it a preferred choice for performance-critical applications.. Acronyms and glossary entries auto-generate at the document's end (defined in `/tail/glossary.typ`).

2.6 Code

Besides writing inline code as such `fn main() { println!("Hello World") }` you can also write code blocks like this:

```rust
fn main() {
  let ship = Starship::new("USS Rustacean", (0.0, 0.0, 0.0));
  let destination = (42.0, 13.0, 7.0);
  let warp = ship.optimal_warp(ship.distance_to(destination));

  println!("🖖 {} traveling to {:?} at Warp {:.2}", ship.name, destination, warp);
  if warp <= 9.0 {
    println!("✨ Warp engaged!");
  } else {
    println!("⚠ Warp failed!");
  }
}
```

Listing 1 - First part of the USS-Rustacean code

or directly from a file

```rust
struct Starship {
  name: String,
  position: (f64, f64, f64),
}

impl Starship {
  fn new(name: &str, position: (f64, f64, f64)) -> Self {
    Self {
      name: name.into(),
      position,
    }
  }
  fn distance_to(&self, dest: (f64, f64, f64)) -> f64 {
    ((dest.0 - self.position.0).powi(2)
      + (dest.1 - self.position.1).powi(2)
      + (dest.2 - self.position.2).powi(2))
    .sqrt()
  }
  fn optimal_warp(&self, distance: f64) -> f64 {
    (distance / 10.0).sqrt().min(9.0)
  }
}
```

Listing 2 - Second part of the USS-Rustacean code from `/resources/code/uss-rustacean.rs`

3 Method

*This chapter defines the project requirements and presents the theoretical background. If applicable, it should also include a **State of the art** review to compare existing solutions and justify the chosen approach.*

Key elements:

- ***Define the requirements** – What must the system/process be able to do?*
- ***Describe the architecture** – What are the key components of the solution?*
- ***Review existing solutions** – What approaches already exist? How does yours compare?*
- ***Explain design choices** – Why did you choose this approach? What alternatives were considered?*

3.1 Requirements

3.1.1 Dataset restrictions

- Due to the high number of tickers, we have only used the crytos with at least 65% of correlation with gold (whatever the method used).
- There is a hole in the dataset downloaded from Kaggle: no data are available between the December 10th 2024 and January 26th 2025. So the data analysis will occur from the first common date with all the interesting tickers and December 9th 2024.

4 Result

This chapter defines the overall system architecture, core components, and interactions between different parts. The goal is to translate the specifications into a concrete, actionable blueprint for development.

Key elements:

- ***Define the architecture*** *– What are the main components of the system, and how do they interact?*
- ***Select technologies*** *– What programming languages, tools, or frameworks will be used? Why?*
- ***Design data structures*** *– How will data be organized and processed?*
- ***Create process flow*** *– What are the key steps or workflows in the system?*
- ***Plan modularity & scalability*** *– How can the design adapt to future needs or extensions?*
- ***Address constraints*** *– What design choices were made due to performance, security, or usability considerations?*

4.1 Dataset

4.1.1 Dataset structure

	date	**ticker**	**close**	**closeNormalized**
index	date	currency short name	price at closing time in $	normalized (by ticker) price

Table 3 - Dataset structure as a Pandas Dataframe

4.1.2 Tickers

The following tickers have more than 80% of correlation (with respect to Pierson's method) with gold over the whole period:

1. PAXG: 0.9997041186031184 correlated with XAU
2. XAUt: 0.9991536823382102 correlated with XAU
3. TRX: 0.914206355226965 correlated with XAU
4. BTC: 0.9043753594253107 correlated with XAU
5. BNB: 0.871339510212942 correlated with XAU
6. LEO: 0.8616999596788716 correlated with XAU
7. GT: 0.8477381081302898 correlated with XAU
8. SUN: 0.847138790223619 correlated with XAU
9. OKB: 0.8155979268149238 correlated with XAU
10. XRP: 0.8146858576249392 correlated with XAU

5 Analysis

This chapter assesses the correctness and performance of the implementation. It includes testing methods, simulations, and any validation techniques used to ensure the system meets its requirements.

Key elements:

- ***Explain verification methods*** *– How do you ensure the system functions correctly?*
- ***Describe validation techniques*** *– How do you prove that the solution meets its objectives?*
- ***Present test results*** *– What experiments, simulations, or benchmarks were conducted?*
- ***Discuss findings*** *– What do the results show? Were there unexpected outcomes?*

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et aut officiis debitis aut rerum necessitatibus saepe eveniet, ut et voluptates repudiandae sint et molestiae non recusandae. Itaque earum rerum defuturum, quas natura non depravata desiderat. Et quem ad me accedis, saluto: 'chaere,' inquam, 'Tite!' lictores, turma omnis chorusque: 'chaere, Tite!' hinc hostis mi Albucius, hinc inimicus. Sed iure Mucius. Ego autem mirari satis non queo unde hoc sit tam insolens domesticarum rerum fastidium. Non est omnino hic docendi locus; sed ita prorsus existimo, neque eum Torquatum, qui hoc primus cognomen invenerit, aut torquem illum hosti detraxisse, ut aliquam ex eo est consecutus? – Laudem et caritatem, quae sunt vitae.

6 Conclusion

This final chapter summarizes the project outcomes, comparing them with the initial objectives. It also reflects on encountered difficulties and discusses possible improvements or future developments.

Key elements:

- ***Summarize the results*** *– What are the key takeaways from the project?*
- ***Compare with objectives*** *– Did the project meet its original goals? Why or why not?*
- ***Reflect on challenges*** *– What were the biggest difficulties, and what was learned?*
- ***Discuss future work*** *– What are possible improvements or next steps?*

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et aut officiis debitis aut rerum necessitatibus saepe eveniet, ut et voluptates repudiandae sint et molestiae non recusandae. Itaque earum rerum defuturum, quas natura non depravata desiderat. Et quem ad me accedis, saluto: 'chaere,' inquam, 'Tite!' lictores, turma omnis chorusque: 'chaere, Tite!' hinc hostis mi Albucius, hinc inimicus. Sed iure Mucius. Ego autem mirari satis non queo unde hoc sit tam insolens domesticarum rerum fastidium. Non est omnino hic docendi locus; sed ita prorsus existimo, neque eum Torquatum, qui hoc primus cognomen invenerit, aut torquem illum hosti detraxisse, ut aliquam ex eo est consecutus? – Laudem et caritatem, quae sunt vitae.

Glossary

Rust **– Rust Programming Language**: Rust is a modern systems programming language focused on safety, speed, and concurrency. It prevents common programming errors such as null pointer dereferencing and data races at compile time, making it a preferred choice for performance-critical applications. 7

IT **– Infotronics** 7

Bibliography

[1] S. Zahno *et al.*, “Dynamic Project Planning with Digital Twin,” *Frontiers in Manufacturing Technology*, vol. 3, May 2023, doi: 10.3389/fmtec.2023.1009633.