Mid term Exam for Financial Econometrics with Python

PRAT Paul; GAVINI Charles; FOURNIER Justin; BLANC Mathieu

November 13, 2024

Contents

1 Introduction **1**

2 Preliminary **1**
2.1 AMAZON . . . 1
2.2 Data Table . . . 1
2.3 Checking the 25 Years range condition . . . 2

3 First Results **2**
3.1 Prices Evolutions . . . 2
3.2 Calculating Returns . . . 3
3.3 Squared Returns . . . 3

4 Amazon and the 8 Stylized Facts **4**
4.0.1 Summary statistics . . . 4
4.1 Prices are non-stationary . . . 4
4.2 Returns are stationary . . . 5
4.3 Asymmetry . . . 5
4.4 Heavy tails . . . 5
4.5 Gaussianity . . . 5
4.5.1 High frequency non-Gaussianity . . . 5
4.5.2 Aggregational Gaussianity . . . 6
4.6 Returns are not autocorrelated . . . 6
4.7 Volatility clustering and long range dependence of squared returns . . . 6
4.8 Leverage effect . . . 6

A Appendix: Python Code **6**

B To go further, CAPM pricing model **6**

1 Introduction

This document provides a comprehensive presentation of our results, including all relevant tables, figures, and calculations. The report is structured into distinct parts, beginning with the importation of essential Python libraries. We then initialize variables to organize the data into different categories (e.g., daily, monthly, returns, log returns), allowing for clear analysis and comparison across various data types and intervals.

2 Preliminary

2.1 AMAZON

The selected stock for this analysis is Amazon due to its significant relevance in current global markets, its impressive growth over time and its position as a major industry leader. The ticker from yahoo finance is "***AMZN*** " on the Nasdaq stock exchange AMAZON on Yahoo Finance First, importing the Amazon stock with yfinance, then display the pandas table. We will import 25 years, 8 months and 25 days of data (from 1999-01-21 to 2024-10-16).

2.2 Data Table

The data printed here is the preview of the Amazon stock extraction from yahoo finance:

Date	Open	High	Low	Close	Adj Close	Volume
1999-01-21	2.612500	2.759375	2.314063	2.650000	2.650000	940964000
1999-01-22	2.487500	3.146875	2.468750	3.075000	3.075000	875316000
1999-01-25	3.037500	3.084375	2.750000	2.809375	2.809375	546476000
1999-01-26	2.815625	3.031250	2.765625	2.877344	2.877344	490696000
1999-01-27	3.353125	3.493750	3.000000	3.140625	3.140625	700452000

Table 1: Preview of Amazon Stock Data from Yahoo Finance

2.3 Checking the 25 Years range condition

We need to verify that the data displays accurately over the 25 years range. Fortunately, the extracted Amazon data has been available since January 1999. To ensure the data's continuity and completeness, we will implement a Python script that identifies and counts any gaps within the dataset. By visualizing the dates of these gaps, we can easily detect any significant interruptions that could potentially impact our data analysis

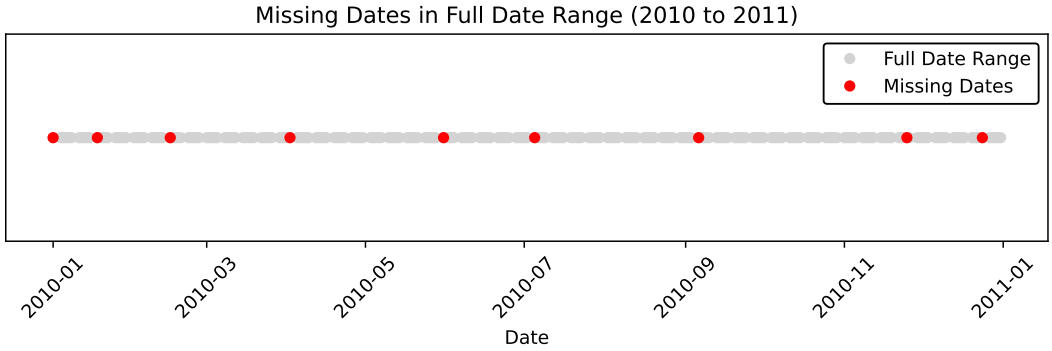

Figure 1: Missing Dates in Full Date Range (2010 to 2011)

We identified a total of 238 isolated days of data gaps per year across the 25-years range (6476 values). Therefore, the data remains reliable for our stylized facts analysis. The missing data points in our dataset are randomly distributed and account for 3.7% of the total data. According to scientific studies on data reliability for volatility testing, a dataset with up to 10 [2]

3 First Results

3.1 Prices Evolutions

With the accuracy and the reliability of our dataset confirmed, we begin by plotting the evolution of prices over 4 different periods : Daily, Weekly, Monthly and Yearly prices.

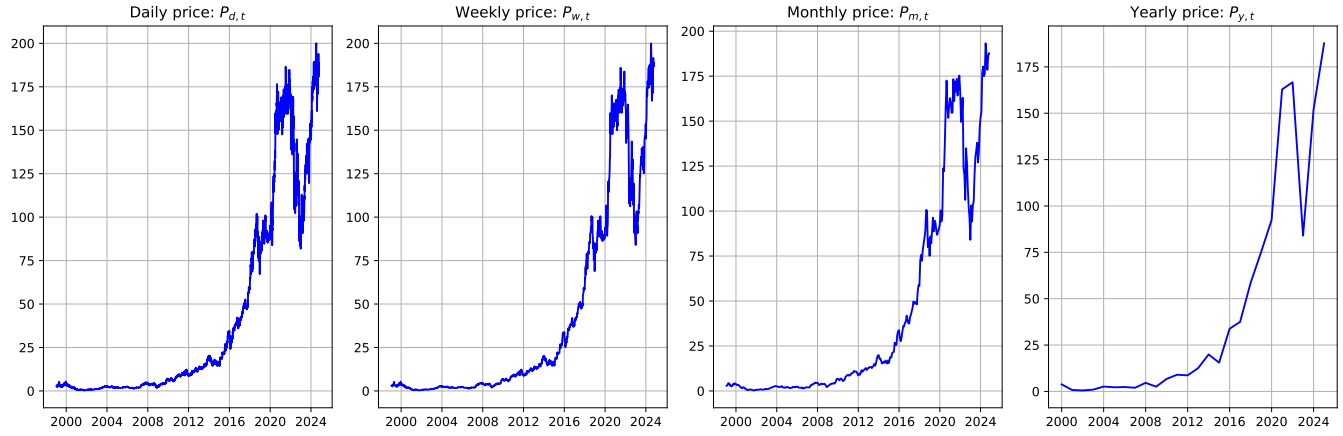

Figure 2: Prices over time by frequency

3.2 Calculating Returns

Using the processed data, we can now output graphs for several key metrics: daily prices, daily log prices, daily simple returns, and daily log returns. Plotting these metrics will allow us to observe daily price movements, the transformation of prices into *log* form for trend analysis, as well as daily returns and their logarithmic equivalents.

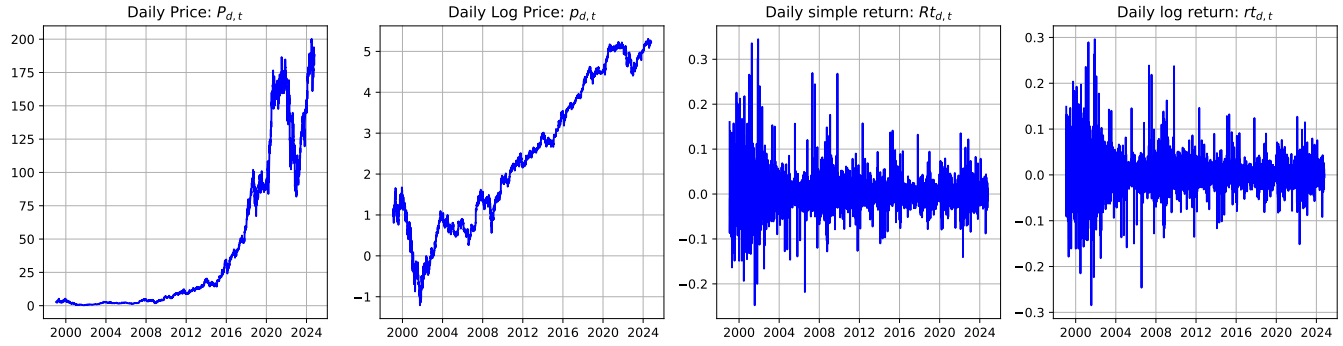

Figure 3: Prices, returns and log returns

3.3 Squared Returns

To complete the analysis of daily price data, we also plot the daily squared returns and daily squared log returns, (providing us a key insight on the volatile behavior on the potential stock risk of our data set.)

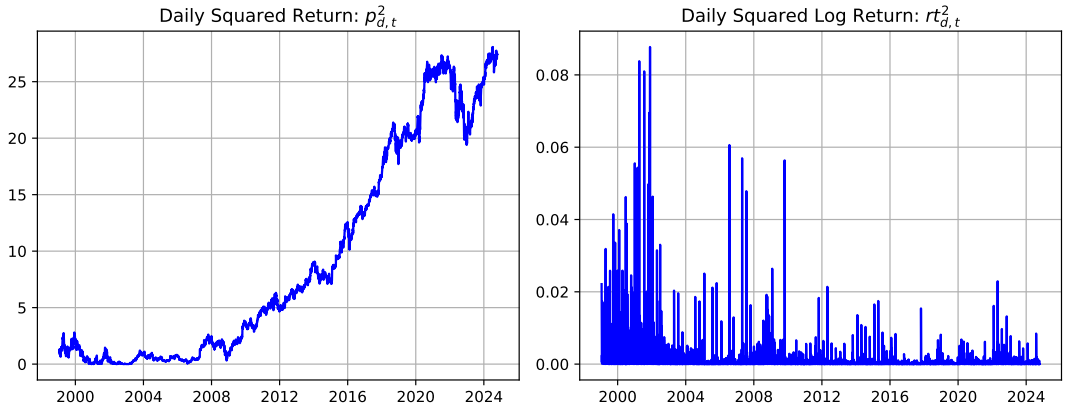

Figure 4: Squared daily returns and daily log returns

4 Amazon and the 8 Stylized Facts

4.0.1 Summary statistics

	daily	weekly	monthly	annual
Mean	0.06351	0.31014	1.32164	22.85692
St.Deviation	3.24131	6.76830	13.06275	45.28052
Diameter.C.I.Mean	0.07896	0.36213	1.45887	18.11598
Skewness	0.41992	0.05008	-0.45895	-0.15137
Kurtosis	11.15158	7.60655	2.59462	-0.64793
Excess.Kurtosis	8.15158	4.60655	-0.40538	-3.64793
Min	-28.45678	-38.51804	-53.02674	-68.54809
Quant5	-4.61051	-9.74288	-20.16713	-55.72274
Quant25	-1.25994	-2.64062	-4.98163	-7.23999
Median	0.04108	0.30519	2.09626	23.07665
Quant75	1.39659	3.40897	8.45973	55.96192
Quant95	4.47118	10.67416	20.90661	94.77653
Max	29.61811	56.11507	48.35221	102.44636
Jarque.Bera.stat	33735.75720	3235.87866	97.20696	0.51147
Jarque.Bera.pvalue.X100	0.00000	0.00000	0.00000	77.43487
Lillie.test.stat	0.10194	0.09591	0.08194	0.06494
Lillie.test.pvalue.X100	0.10000	0.10000	0.10000	99.00000
N.obs	6474.00000	1342.00000	308.00000	24.00000

Table 2: Summary statistics for the amazon stock

4.1 Prices are non-stationary

The first feature that will highlight non-stationarity of the prices is the comparison of p_t vs p_{t-1}.

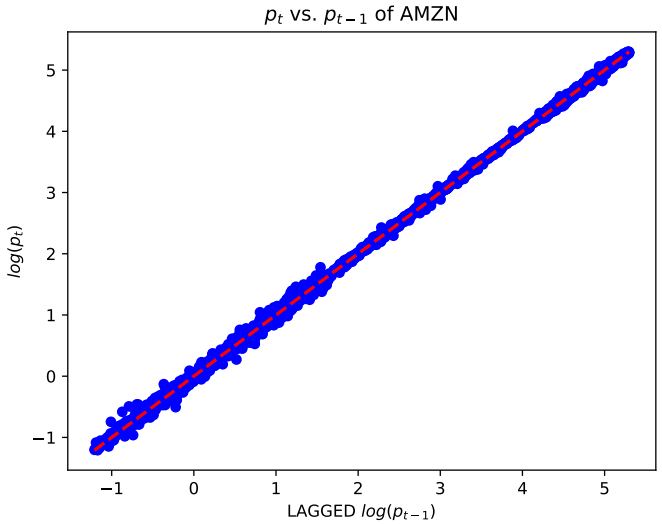

Figure 5: Comparison of $\log(p_t)$ vs $\log(p_{t-1})$

The graph in Figure 5 demonstrates this strong linear relationship, indicating that Amazon's prices at time t are highly dependent on those at $t-1$ and lack mean reversion, supporting the idea of non-stationarity.
Additionally, the empirical autocorrelation function (ACF) of Amazon's daily prices shows a slow decay, further suggesting non-stationarity, as shown in the next figure.

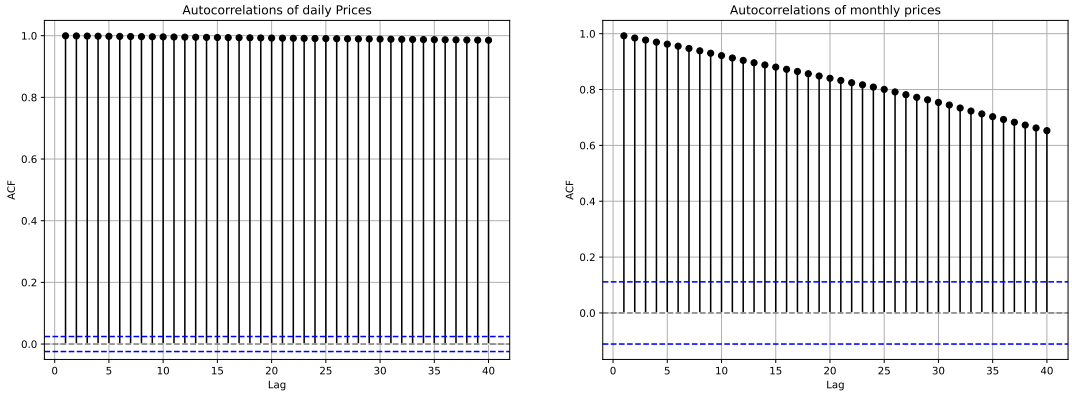

Figure 6: Autocorrelations of daily and monthly Prices (1999-2024)

4.2 Returns are stationary

TODO

4.3 Asymmetry

TODO **Stylized Fact 3: The distribution of returns is asymmetric and often negatively skewed**, reflecting the fact that the downturns of financial markets are often much steeper than the recoveries. Investors tend to react more strongly to negative news than to positive news.

	daily	weekly	monthly	annual
Skewness	0.41992	0.05008	-0.45895	-0.15137
Kurtosis	11.15158	7.60655	2.59462	-0.64793

Table 3: Skewness and kurtosis for *log* returns

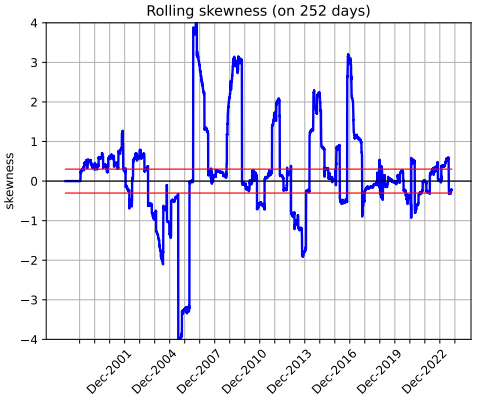

Figure 7: Rolling skewness

4.4 Heavy tails

TODO

4.5 Gaussianity

TODO

4.5.1 High frequency non-Gaussianity

As the result in Table 2 the skewness is positive for daily and monthly data [1] TODO add an article about amzn skewness. The 3rd central moment is defined as $\mu_3 := E((X - m_1)^3)$. The skewness of r_t is defined as:

$$\text{Skew}(r_t) := E\left[\left(\frac{X - m_1}{\sigma}\right)^3\right] = \frac{\mu_3}{\sigma^3} = \frac{\mu_3}{\mu_2^{3/2}}.$$

In this case, and as emphasized in the 3^{rd} stylized fact, $\text{Skew}(r_t) > 0$, large realizations of X are more often larger than the mean μ. Skewness is thus used as a measure of asymmetry of the distribution $f_X(x)$. Therefore:

$\text{Skew}(r_t) > 0$, so the distribution is said to be **right skewed**.

$\text{Skew}(r_t) > 0$, then $\mu >$ median, where the median is the 50% quantile of the distribution.

4.5.2 Aggregational Gaussianity

TODO

4.6 Returns are not autocorrelated

TODO

4.7 Volatility clustering and long range dependence of squared returns

TODO

4.8 Leverage effect

TODO

A Appendix: Python Code

Below is the Python code used in this analysis.

```python
# Python code example
import numpy as np
import pandas as pd

def analyze_data(data):
    mean = np.mean(data)
    std_dev = np.std(data)
    return mean, std_dev

data = [1, 2, 3, 4, 5]
mean, std_dev = analyze_data(data)
print(f"Mean: {mean}, Standard Deviation: {std_dev}")
```

Listing 1: Python Code for Analysis

B To go further, CAPM pricing model

References

[1] John Doe and Jane Smith. An example article. *Journal of Examples*, 42(1):1–10, 2023.

[2] Giovanni Pumi et al. Estimation of long-range dependent models with missing data: to impute or not to impute? *arXiv preprint*, 2023.