SA2-Part 2

Espiritu, Joseph Raphael M. Clores, Harneyyer Leosara

2025-05-17

1. Data Gathering and Info

- **Kaggle Data** (Citation)
  - author = {Prasoon Kottarathil},
  - title = {Ethereum Historical Dataset},
  - year = {2020},
  - publisher = {kaggle},
  - journal = {Kaggle Dataset},
  - how published = { https://www.kaggle.com/prasoonkottarathil/ethereum-historical-dataset}
  - Columns are **Unix Timestamp**, **Date**, **Symbol**, **Open**, **High**, **Low**, **Close**, **Volume**

```
library(reticulate)

# Create and activate the virtual environment
virtualenv_create("r-reticulate-env")
```

```
## virtualenv: r-reticulate-env
```

```
virtualenv_install("r-reticulate-env", packages = c("pandas", "numpy", "matplotlib", "scipy"))
```

```
## Using virtual environment "r-reticulate-env" ...
```

```
## + "C:/Users/josep/OneDrive/Documents/.virtualenvs/r-reticulate-env/Scripts/python.exe" -m pip instal
```

```
use_virtualenv("r-reticulate-env", required = TRUE)
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import shapiro

# Load the Ethereum data
df = pd.read_csv("ETH_1H.csv")

# Sort by date and preprocess
df = df.sort_values("Date")
df["Date"] = pd.to_datetime(df["Date"])
df["Close"] = pd.to_numeric(df["Close"], errors='coerce')
df = df.dropna(subset=["Close"])

# Calculate log returns
df["log_return"] = np.log(df["Close"] / df["Close"].shift(1))
df = df.dropna(subset=["log_return"])

# Sample 5000 log returns
sample_size = 5000
```

```
sampled_returns = df["log_return"].sample(n=sample_size, random_state=42)

# Shapiro-Wilk Test (on sample)
stat, p = shapiro(sampled_returns)

print("Shapiro-Wilk Test Results (on sample of 5000):")
```

```
## Shapiro-Wilk Test Results (on sample of 5000):
```

```
print(f"Test Statistic = {stat:.4f}")
```

```
## Test Statistic = 0.8097
```

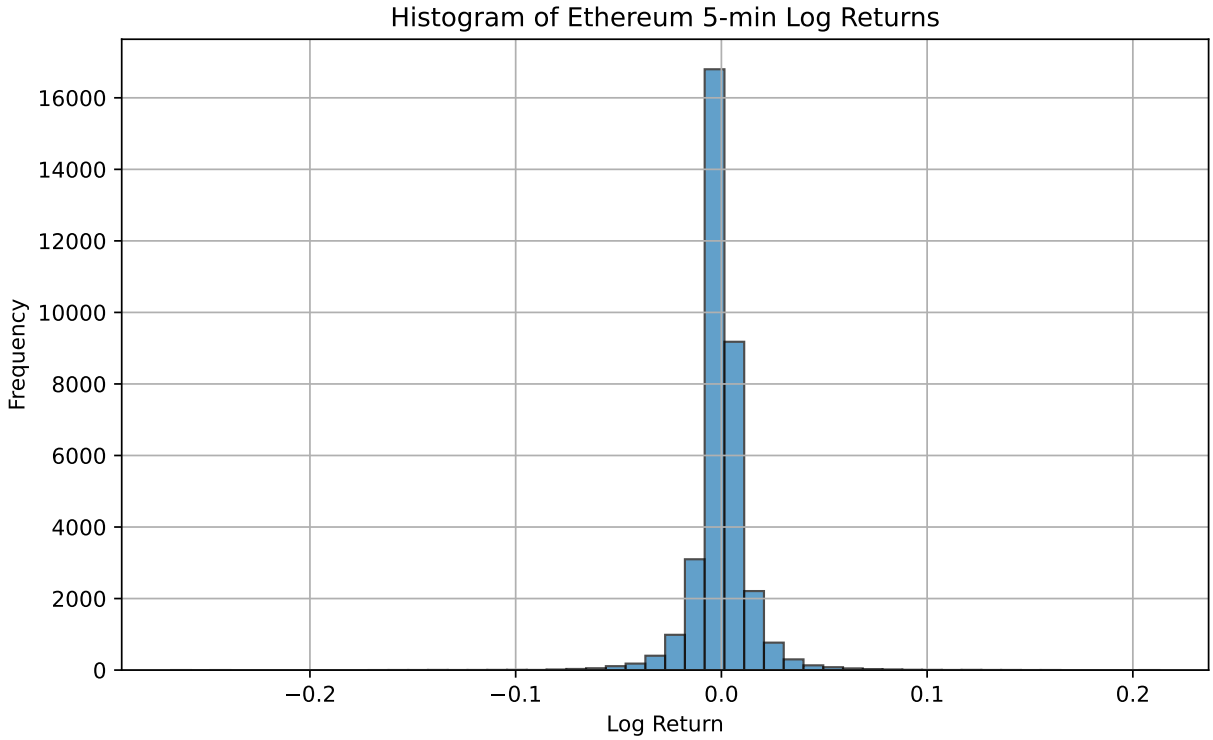
```
print(f"p-value = {p:.4f}")
```

```
## p-value = 0.0000
```

```
if p > 0.05:
    print("The data is likely normally distributed (fail to reject H0).")
else:
    print("The data is NOT normally distributed (reject H0).")
```

```
## The data is NOT normally distributed (reject H0).
```

```
# Histogram of log returns
plt.figure(figsize=(8,5))
plt.hist(df["log_return"], bins=50, edgecolor='black', alpha=0.7)
plt.title("Histogram of Ethereum 5-min Log Returns")
plt.xlabel("Log Return")
plt.ylabel("Frequency")
plt.grid(True)
plt.tight_layout()
plt.show()
```



2. Interpretation and Results -Interpretation of Shapiro-Wilk Test on Ethereum 5-min Log Returns

- The **Shapiro-Wilk test** was applied to a **random sample of 5,000 log returns** from the Ethereum dataset.
- **Test Statistic:** around `0.8` to 0.81
  - Values **closer to 1** indicate data is more likely to be normally distributed.
- **p-value:** `< 0.0000`
  - Since $\mathbf{p < 0.05}$, we **reject the null hypothesis** that the log returns follow a normal distribution.
- The **5-minute log returns** of Ethereum **do not follow a normal distribution**.
- This implies that the returns exhibit **non-normal characteristics**, commonly seen in financial data:
  - Likely **heavy tails**, **skewness**, or **volatility clustering**.
- Models that assume normal returns may **underestimate risk** and fail to capture real-world price behavior.
- Consider using alternative models that better fit