1 Methodology

1.1 Data Collection

1.1.1 Introduction

This document provides a detailed description of the Coinbase Bitcoin dataset. The dataset consists of the daily closing prices of Bitcoin in U.S. Dollars from Coinbase, not seasonally adjusted, with data starting from December 1, 2014, up to June 23, 2024. The data is recorded daily at 5 PM PST and is provided with a 7-day frequency.

1.1.2 Broad Objective

The broad objective of this dataset documentation is to provide researchers, analysts, and enthusiasts with comprehensive information about the Coinbase Bitcoin dataset. This includes its source, format, contents, and potential applications in financial analysis, econometrics, and cryptocurrency research.

1.1.3 Purpose of Data Collection

The Coinbase Bitcoin dataset was collected to facilitate comprehensive analysis and understanding of the daily closing prices of Bitcoin in U.S. Dollars over a significant period. The primary objectives of collecting this data include:

- **Time Series Analysis:** To analyze and model the trends, patterns, and volatility exhibited by Bitcoin prices over time.
- **Statistical Modeling:** To develop statistical models that can capture the stochastic nature of Bitcoin price movements, aiding in forecasting future prices.
- **Market Behavior Studies:** To study the behavior of the Bitcoin market, including identifying regime shifts and exploring correlations with external factors.
- **Investment Analysis:** To provide insights for investors and stakeholders interested in Bitcoin as an investment asset, helping them make informed decisions.
- **Research and Education:** To support academic research and educational initiatives focused on cryptocurrencies, financial markets, and quantitative finance.

1.1.4 Data Sources

The following table summarizes the data source and its associated web link used to compile the dataset:

Data Source	Description
Coinbase Bitcoin (CBBTCUSD) on FRED	2014-2024

Table 1: List of data sources used to construct the dataset.

1.1.5 Data Description

The description of headers/column-names of the constructed dataset is given in the table below:

Attributes	Description	Data type
Date	The date of the recorded Bitcoin price	Character
CBBTCUSD	Closing price of Bitcoin in U.S. Dollars at 5 PM PST	Character

Table 2: Description of each header of the constructed dataset.

1.1.6 Data file format

This dataset is stored in the following file formats:

- CSV: Coinbase_Bitcoin_Data.csv

A random subset of size 5 is selected from the constructed dataset and printed below for clarity.

Date	Price
2015-01-01	320.53
2016-05-15	457.34
2017-12-31	13377.60
2019-07-10	12193.99
2022-11-20	16604.48

Table 3: A random subset of the constructed dataset.

1.1.7 Additional Information

- **Units:** U.S. Dollars, Not Seasonally Adjusted
- **Frequency:** Daily, 7-Day
- **Data Source:** Coinbase
- **Copyright:**

1.2 Data Preprocessing

Preprocessing the dataset involved the following steps:

- **Loading and Inspecting the Data:**

  Initially, the data was loaded and inspected to understand its structure and the types of values it contained.

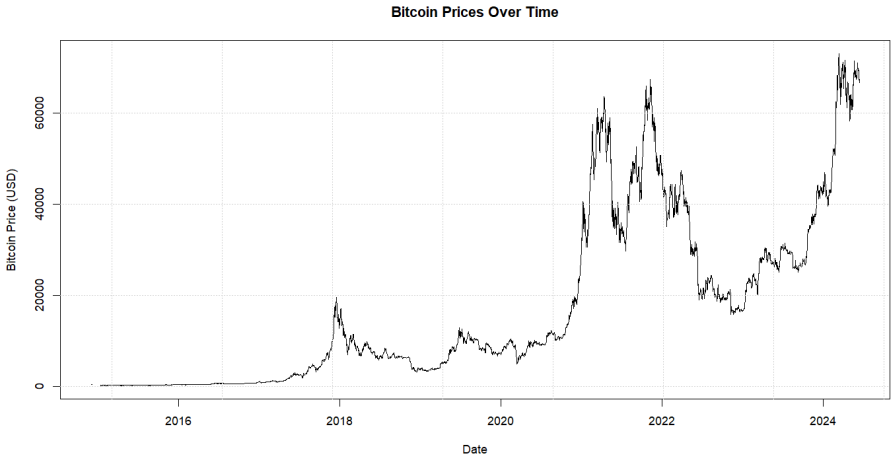

Figure 1: Time Series Plot of Bitcoin Closing Prices from December 1, 2014, to June 23, 2024

- **Handling Missing Values:**

  Missing values in the dataset were identified and replaced using linear interpolation to ensure a continuous time series without gaps. There were 35 missing values, which were filled using the interpolation technique.Additionally, the data types of the 'Date' and 'Bitcoin' columns, initially in character form, were converted to numerical for statistical analysis and model fitting.

  For interpolation, the `na.approx` function from the `zoo` package was used with `rule=2`. This setting ensures that missing values are linearly interpolated based on existing data points, preserving the overall trend of the time series.

  The function call `na.approx(df$Bitcoin, rule=2)` performs linear interpolation with specific rules governing the handling of missing values at the boundaries:

  - **rule=1:** NA values are not extrapolated and remain NA at the boundaries if they cannot be interpolated.

- **rule=2:** Linear interpolation is performed, and if necessary, the closest non-NA value is carried forward or backward to fill NA values at the boundaries.
- **rule=3:** NA values are extrapolated using a linear model based on the closest available values, extending the trend beyond the existing data points.
- **rule=4:** This rule is similar to rule=2 but ensures that the NA values at the beginning or end are replaced with the mean of the interpolated values, providing a balanced approach for boundary values.

Using `rule=2` ensures a practical approach where missing values are filled using the closest available data, maintaining the integrity of the time series for further analysis.

- **Transformation:**

  The Bitcoin values were transformed using the natural logarithm to stabilize the variance and make the data more suitable for modeling.

  The logarithm transformation (log(Bitcoin)) is applied to manage the large range of Bitcoin values, simplifying numerical computations and stabilizing variance for statistical analysis and modeling purposes and plotted the data plot using ggplot library.

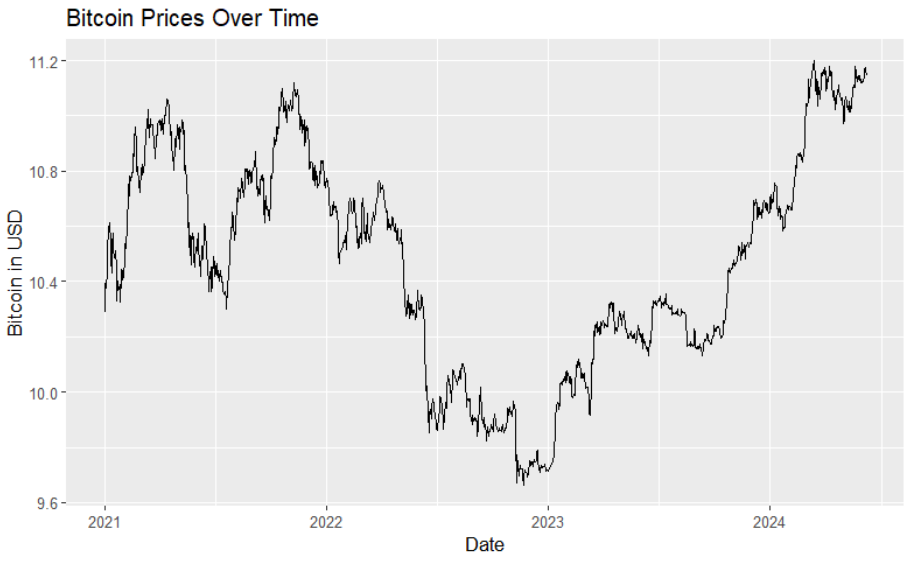

Figure 2: Time Series Plot of Bitcoin Closing Prices from January 1, 2021, to June 7, 2024

- **Subsetting the Data:** The dataset was resized to focus on the period from 2021 to 2024. This was done to leverage recent data which exhibited

more significant fluctuations and was expected to provide better predictive insights.

1.3 Exploratory Data Analysis (EDA)

EDA was conducted to understand the statistical properties and underlying patterns in the data:

- **Summary Statistics:** Descriptive statistics such as mean, variance, and standard deviation were computed.

  Mean of the Bitcoin column of the Dataset is: 10.46837
  Variance of the Bitcoin column of the Dataset is: 0.1590694
  Standard Deviation of the Bitcoin column of the Dataset is: 0.3988351

- **Visualization:** Time series plots, along with Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots, were generated to visualize the data and identify any temporal dependencies.

  ]

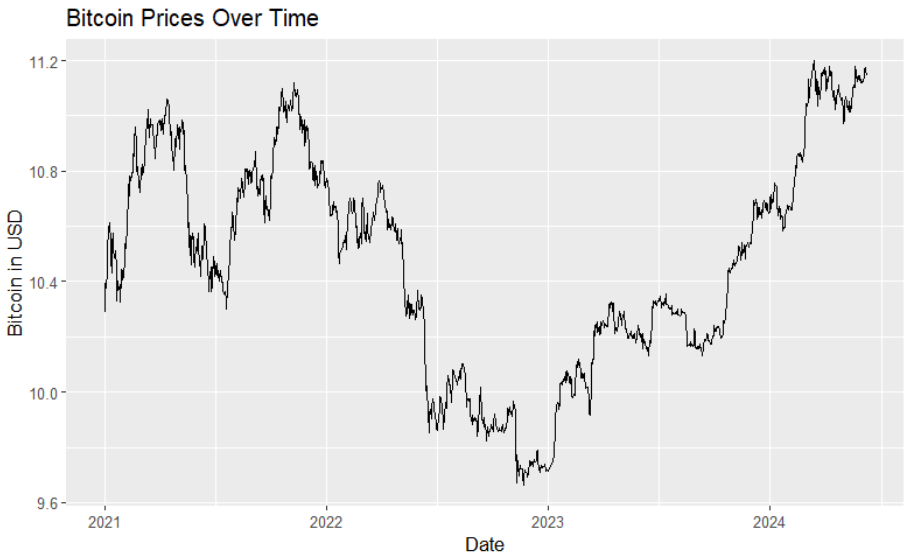

Figure 3: Time Series Plot of Bitcoin Closing Prices from January 1, 2021, to June 7, 2024

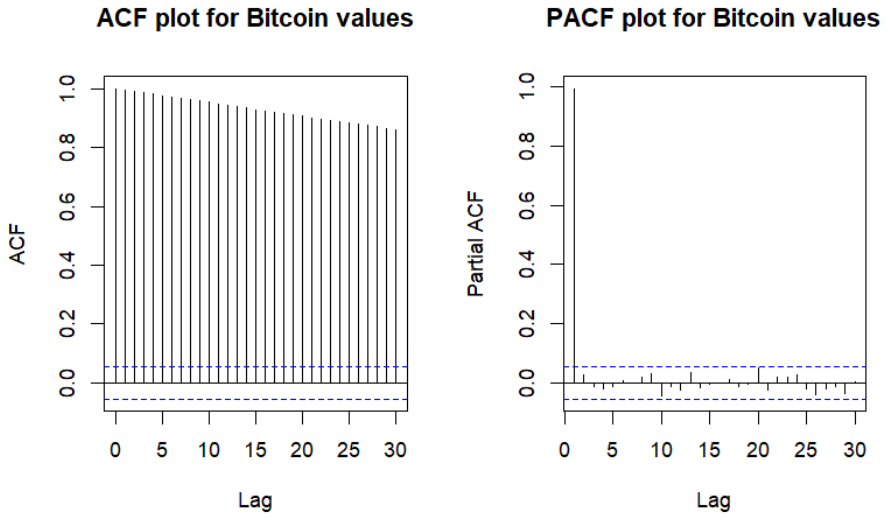

Figure 4: ACF and PACF Plots of Bitcoin Closing Prices from January 1, 2021, to June 7, 2024

1. Plot ACF and PACF for your data.
2. Identify significant lags in ACF exceeding confidence intervals.
3. Identify significant lags in PACF exceeding confidence intervals.
4. Analyze ACF for gradual decay (AR component) or sudden drop (MA component).
5. Analyze PACF for sudden drop (AR order) or gradual decay (MA component).
6. Determine MA order (q) from ACF and AR order (p) from PACF.

1.4 Modeling and Forecasting

Three different models were considered for the time series analysis and forecasting:

1.4.1 ARIMA Model

- **ARIMA Model Summary:** The ARIMA model with parameters AR(1), differencing of order 1, and MA(0) is represented by the following equation:

$$(1 - B)(1 - B^1)Y_t = c + \epsilon_t$$

where:

- Y_t is the value of the time series at time t,
- B is the backshift operator ($BY_t = Y_{t-1}$),
- $(1 - B)$ and $(1 - B^1)$ are the differencing operators,
- c is the constant term (intercept),
- ϵ_t is the error term at time t, assumed to be independently and identically distributed (i.i.d.) with mean zero and constant variance.

The ARIMA model parameters are denoted as ARIMA(1, 1, 0):

- $p = 1$: Number of autoregressive (AR) terms, capturing the effect of one previous value on the current value.
- $d = 1$: Degree of differencing, transforming the time series into a stationary series.
- $q = 0$: Number of moving average (MA) terms, assuming no moving average component.

The ARIMA model integrates autoregressive (AR), differencing (I), and moving average (MA) components to model and forecast time series data, accommodating trends and autocorrelation structures effectively.

- **Model Fitting:** The ARIMA(1, 1, 0) model was fitted to the transformed Bitcoin data after 1st order differencing.

  We use auto.arima function if from acf and pacf plot we are not sure about the order or AR and MA so that time auto.arima() is very helpful. After fitting the model, the forecasting equation obtained is

$$\hat{y}_t = -0.0364 \times y_t$$

  where $\hat{y}_t$ represents the predicted values based on the model and y_t denotes the transformed Bitcoin data after 1st order differencing.

  This linear equation suggests a negative relationship between the transformed Bitcoin values and the predicted outcomes, indicating that as the Bitcoin values increase (or decrease), the model predicts a corresponding decrease (or increase).

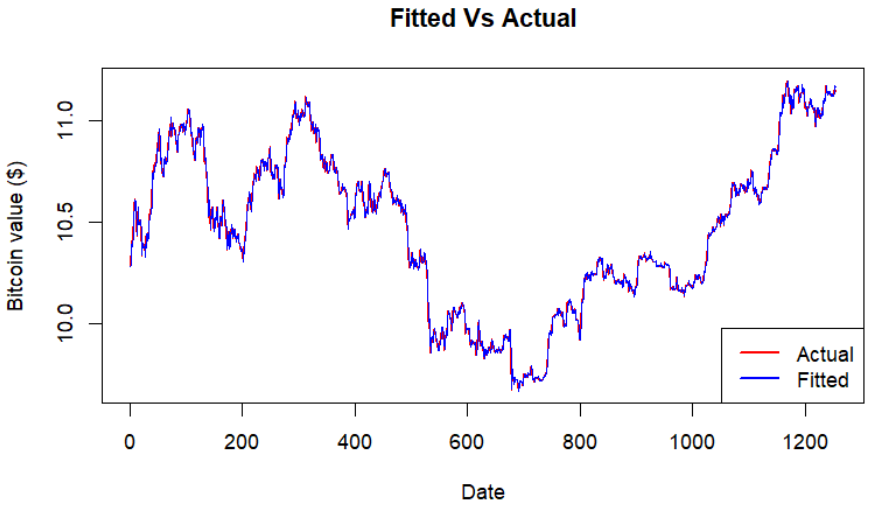

Figure 5: Model Fitting

- **Residual Analysis:** The residuals of the fitted model were analyzed using diagnostic plots to ensure they behaved like white noise, indicating a good fit.

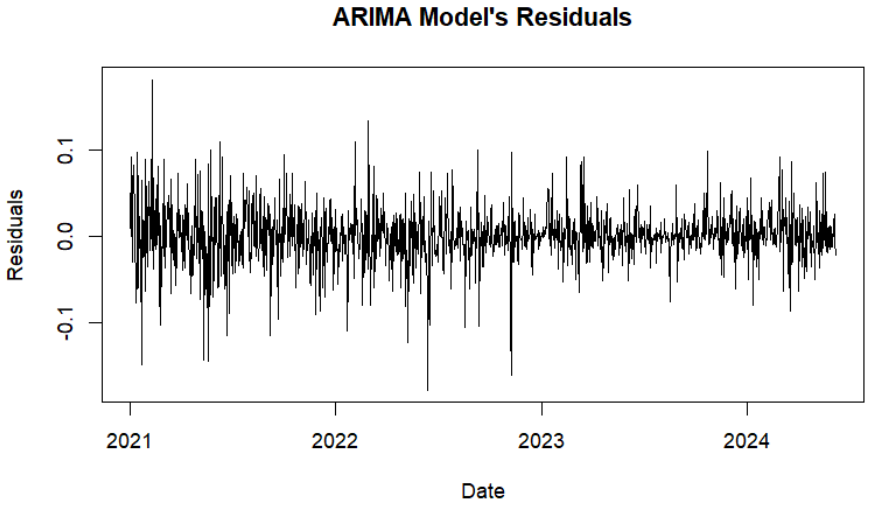

Figure 6: Residuals Plots of ARIMA model

1.4.2 Threshold Autoregressive (TAR) Model for 2 and 3 Regime

- **Model Specification:** The Threshold Autoregressive (TAR) model is a nonlinear time series model that captures regime shifts or nonlinear behaviors in the data. It is particularly suited for situations where the data exhibit distinct periods of behavior or different response patterns based on past observations.

- **Model Equation:**

  The TAR model used in this analysis can be described as follows:

$$y_t = \begin{cases} \mu_1 + \phi_1 y_{t-1} + \epsilon_t & \text{if } y_{t-1} \leq \theta \\ \mu_2 + \phi_2 y_{t-1} + \epsilon_t & \text{if } y_{t-1} > \theta \end{cases}$$

  Where:

  - y_t represents the Bitcoin price at time t.
  - μ_1 and μ_2 are intercepts for the lower and upper regimes, respectively.
  - ϕ_1 and ϕ_2 denote autoregressive coefficients associated with the lower and upper regimes.
  - θ denotes the threshold value that determines the shift between regimes based on the lagged value y_{t-1}.
  - ϵ_t represents the error term assumed to be white noise.

- **Model Fitting for 2 regime Tar:**

  - **The fitted model:** The TAR model was fitted to the Bitcoin price data using the `setar` function from the `tsDyn` package in R:

$$\hat{y}_t = \begin{cases} 0.0256225 + 0.9976059 \cdot y_t, & \text{if } y_{t-1} \leq 10.84 \\ 0.5247176 + 0.9523690 \cdot y_t, & \text{if } y_{t-1} > 10.84 \end{cases}$$

  - **Model Fitting:** The TAR model with two regimes was fitted to the data using a threshold of 10.471078:

    The TAR model is defined by the following equations:

    * **Regime 1:** $\hat{y}_t = 0.0256225 + 0.9976059 \cdot y_t$
    * **Regime 2:** $\hat{y}_t = 0.5247176 + 0.9523690 \cdot y_t$

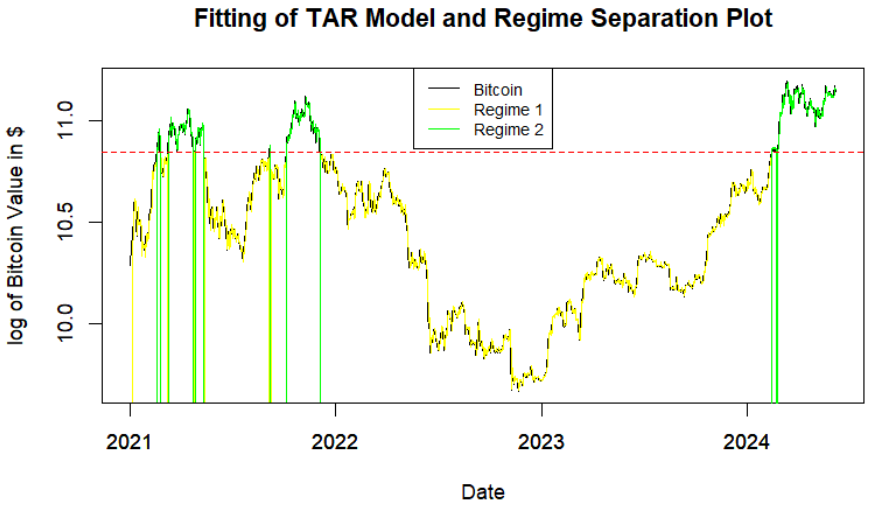

Figure 7: Model Fitting of TAR Model

- **Residual Analysis:** Residuals of the TAR model were examined to ensure they met the assumptions of the model.

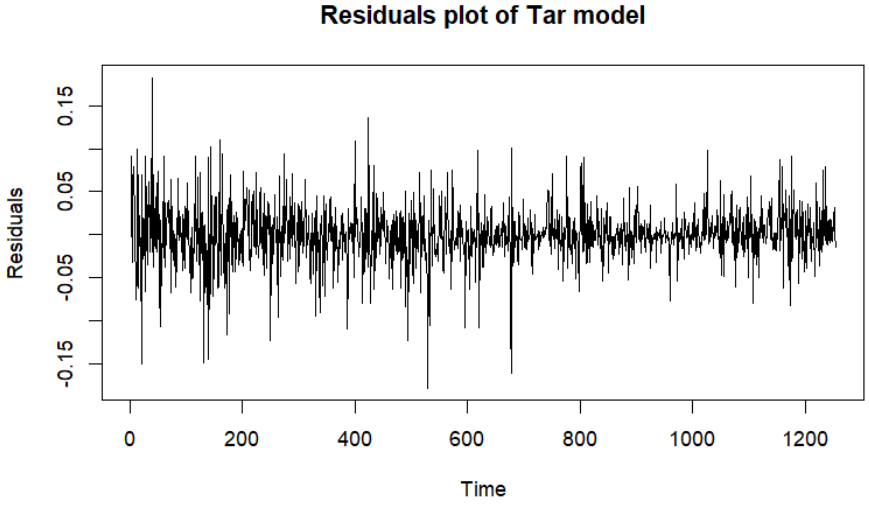

Figure 8: Residuals of TAR Model

- **Regime Plotting:** The TAR model with two regimes effectively captures

the different phases of Bitcoin value dynamics. The regime separation helps in understanding the behavior of Bitcoin values in different conditions, with the threshold acting as a critical point for switching regimes. The frequent regime changes highlight the high volatility of Bitcoin values, especially around the threshold level.

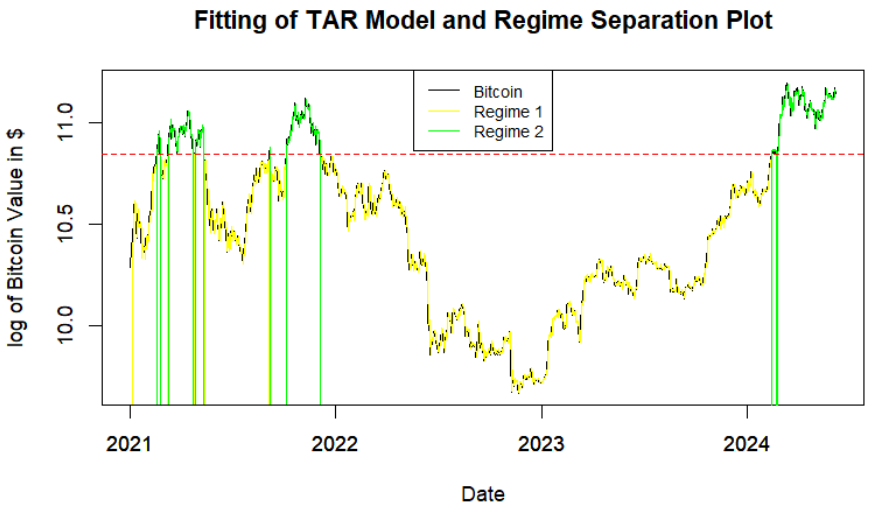

Figure 9: Regime Plot of TAR Model

- The Bitcoin value shows significant volatility, switching between regimes frequently in some periods (e.g., 2021 and early 2022).
- More stable periods are also visible, such as in late 2022 and 2023, where the value remains mostly within one regime.
- The TAR model helps in identifying different behaviors of the Bitcoin value based on its level relative to the threshold.
- For instance, during periods where the value is in Regime 2 (above the threshold), the value generally trends upward or remains stable.
- In contrast, Regime 1 (below the threshold) often corresponds to downward trends or increased volatility.