FOSSEE Summer Fellowship Report

on

FLOSS - R

Submitted by

Arpit Dubey
Indian Institute of Technology, Kharagpur

under the guidance of

Prof. Radhendushka Srivastava
Mathematics Department,
IIT BOMBAY

Prof. Kannan M. Moudgalya
Chemical Engineering Department,
IIT BOMBAY

and supervision of

Debatosh Chakraborty
Project Research Associate,
R Team, FOSSEE,
IIT Bombay

July 2024

Acknowledgment

I would like to express my sincere gratitude to Prof. Kannan M. Moudgalya, Department of Chemical Engineering, IIT Bombay, for creating the FOSSEE Fellowship programme and providing students from all over India with the opportunity to participate in it. I would equally like to thank my FLOSS mentor, Prof. Radhendushka Srivastava, Department of Mathematics, IIT Bombay, for his immense support and knowledge throughout this research project and for helping me with various concepts. I would also like to express my gratitude to other members of the R team, especially my mentor, Mr. Debatosh Chakraborty, for the guidance and valuable input throughout the fellowship. I am very grateful to have been given such a fantastic opportunity to work on this exciting project.

Contents

Chapter 1

Introduction

This report contains all the contributions made by me during the FOSSEE Summer Fellowship 2024 from 15th May 2024 to 15th July 2024. I did the fellowship under the guidance of the R team under the FOSSEE Project. My contributions include a Case Study Project on Analysis of Bitcoin Data: Modeling and Forecasting, TBC Project (Textbook Companion Project), and Creation of Material on Linear Time Series Models.

The FOSSEE (Free/Libre and Open Source Software for Education) project promotes the use of FLOSS tools to improve the quality of education in our country. The project aims to reduce dependency on proprietary software in educational institutions. The FOSSEE project is part of the National Mission on Education through Information and Communication Technology (ICT), Ministry of Education (MoE), Government of India

Chapter 2

Contribution to the TextBook Companion (TBC) project

As a part of the selection procedure for the FOSSEE Summer Fellowship, an applicant is required to select a standard textbook related to Probability, Statistics, Algebra, etc., with at least 80 solved examples to submit a TBC proposal for the R TBC project. My proposal got approved, and during the fellowship period, I contributed to the R TBC project by creating a R textbook companion for the below-mentioned textbook:

Textbook Name	**Author**	**Edition**
John E. Freund's Mathematical Statistics with Applications	Pearson Education Limited	8th

Table 2.1: Details of the textbook selected for R TBC contribution.

I have coded 170 solved problems of the book. The R codes for the Textbook Companion can be accessed here.

Chapter 3

Analysis of Bitcoin Data: A Case Study on Modeling and Forecasting

3.1 Data Collection

3.1.1 Introduction

The dataset contains daily closing prices of Bitcoin in USD from Coinbase, spanning December 1, 2014, to June 23, 2024. It is valuable for financial analysis, econometrics, market behavior studies, investment analysis, and educational research on Bitcoin market dynamics.

3.1.2 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 3.1: List of data sources used to construct the dataset.

3.1.3 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 3.2: Description of each header of the constructed dataset.

3.2 Data Analysis

Preprocessing the dataset involved the following steps:

- **Loading the Data:**

```r
path = "CBBTCUSD.csv"
df = read.csv(path)
head(df)
```

Description: df [6 × 2]

	DATE <chr>	CBBTCUSD <chr>
1	2014-12-01	370
2	2014-12-02	378
3	2014-12-03	378
4	2014-12-04	377.1
5	2014-12-05	.
6	2014-12-06	378

6 rows

Figure 3.1: First 6 values of the dataset

The first six values of the dataset reveal some missing values. Sometimes, the dataset may not have any missing values at the beginning, so we should check for missing values using the sum(is.na(data)) function every time we perform analysis.

- **Inspecting the Data:**

```r
str(df)
```

```
'data.frame':   3483 obs. of  2 variables:
 $ DATE    : chr  "2014-12-01" "2014-12-02" "2014-12-03" "2014-12-04" ...
 $ CBBTCUSD: chr  "370" "378" "378" "377.1" ...
```

Figure 3.2: summary of the Data

- **Plotting of the Data:**

```r
plot(as.Date(df$DATE), as.double(df$CBBTCUSD), type = "l",xlab =
    "Date", ylab = "Bitcoin Price (USD)",main = "Bitcoin Prices
    Over Time")
```

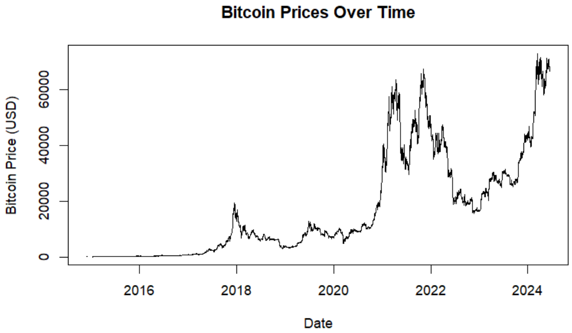


Figure 3.3: Time Series Plot of Bitcoin Closing Prices

- **Columns datatype, renaming, and handling missing values:**

```r
# Assign column names
colnames(df)= c("Date","Bitcoin")
# Convert the datatype
df$Date=as.Date(as.character(df$Date))
df$Bitcoin=as.double(as.character(df$Bitcoin))
# Number of missing entries
cat("total number of missing values in tha entire Dataset is :",
    sum(is.na(df)))
```

```
Total number of missing values in tha entire Dataset is : 35
```

Figure 3.4: Number of missing values

- **Handling Missing Values:** As we have seen, there are missing values in our dataset. There are many ways to fill these missing values, but since the dataset is continuous, interpolation is a better method to address the missing values.

```r
df$Bitcoin <- na.approx(df$Bitcoin, rule=2)
```

For Linear interpolation, the `na.approx` function was used with `rule=2`.

- **Subsetting the Data:**The dataset was resized to focus on the period from 2021 to 2024. This was done to use new data to make better predictions.

```r
df <- df[df$Date >= as.Date("2021-01-01"), ]
```

- **Transformation:**

  We have applied a logarithmic transformation to the Bitcoin column, as the log transformation preserves the proportional relationships between data points.

```
df$Bitcoin=log(df$Bitcoin)
ggplot(dframe, aes(x = Date, y = Bitcoin)) +geom_line()+xlab("
    Date")+ylab("Bitcoin in USD")+labs(title = "Bitcoin Prices
    Over Time")
```

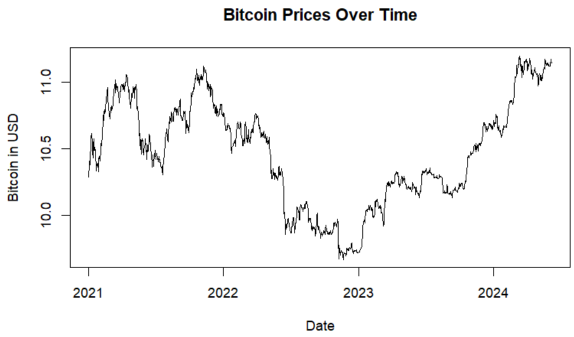


Figure 3.5: Plot of Bitcoin Closing Prices

- **Visualization:** Time series plots (see Figure 3.5), along with Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots, have been generated to visualize the data and identify any temporal dependencies.

```
ggplot(dframe, aes(x = Date, y = Bitcoin)) +geom_line()+xlab("
    Date")+ylab("Bitcoin in USD")+labs(title = "Bitcoin Prices
    Over Time")
acf(dframe$Bitcoin,main="ACF plot for Bitcoin values")
pacf(dframe$Bitcoin,main="PACF plot for Bitcoin values")
```

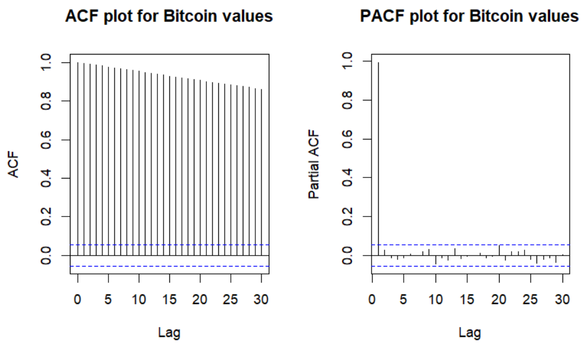


Figure 3.6: ACF and PACF Plots

1. Plot ACF and PACF for your data.
2. Identify significant lags in ACF/PACF exceeding confidence intervals.

3.3 Modeling and Forecasting

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

3.3.1 ARIMA Model

- **ARIMA Model Summary:** We used auto.arima(), which provides the values of the parameters for the ARIMA model (i.e., ARIMA(p,d,q)) that best fit the data. Here:
  - Model includes 1 autoregressive term, 1 differencing term for stationarity, and no moving average component.
- **Model Fitting:** The ARIMA(1, 1, 0) model was fitted to the transformed Bitcoin data with 1st order differencing.

```
arima_model=auto.arima(dframe$Bitcoin)
summary(arima_model)
arima_fitted_val= fitted(arima_model)
plot(dframe$Bitcoin, col = 'red', lwd = 1, main = "Fitted Vs Actual", ylab = "Bitcoin value ($)",xlab="Date", type = "l")
lines(arima_fitted_val, col = "blue", lwd = 1)
legend('bottomright', legend = c("Actual", "Fitted"), col = c("red", "blue"), lwd = 2)
```

```
Series: dframe$Bitcoin
ARIMA(1,1,0)

Coefficients:
          ar1
      -0.0364
s.e.   0.0283

sigma^2 = 0.001116:  log likelihood = 2481.58
AIC=-4959.16   AICc=-4959.15   BIC=-4948.89

Training set error measures:
                       ME       RMSE        MAE        MPE      MAPE     MASE         ACF1
Training set 0.0007174924 0.03337957 0.02287173 0.00617688 0.2179083 0.996358 0.0009469197
```

Figure 3.7: Arima Model Summary

auto.arima() simplifies model selection by automatically identifying and fitting the most suitable ARIMA model based on the characteristics of the data. After fitting the model, the forecasting equation obtained is

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

where $\hat{y}_t$ are the predicted values and y_t the differenced Bitcoin data, showing a negative relationship.

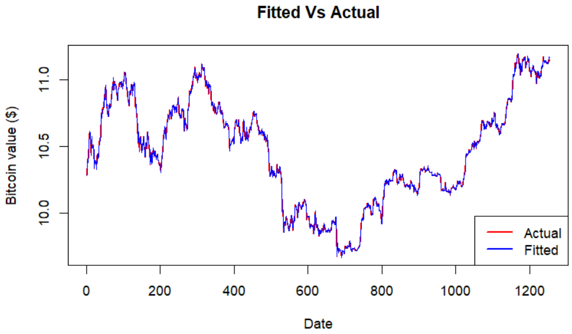


Figure 3.8: Model Fitting

3.3.2 Threshold Autoregressive (TAR) Model for 2 Regime

- **Model Specification:** The Threshold Autoregressive (TAR) model captures regime shifts and nonlinear behaviors in time series data, suited for distinct periods or response patterns.

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

  - **The fitted model:** Now, we will fit the 2 regime TAR model on the Bitcoin data using the `setar` function from the `tsDyn` package in R:

```
tar_model <- setar(dframe$Bitcoin, m = 1, thDelay = 0,
    nthresh = 1, model = "TAR")
summary(tar_model1)
# tar model fitting plot with two regime
const.L <- coef(tar_model)["const.L"]
phiL.1 <- coef(tar_model)["phiL.1"]
const.H <- coef(tar_model)["const.H"]
phiH.1 <- coef(tar_model)["phiH.1"]
threshold <- tar_model$coefficients["th"]
regime1 <- numeric(length(dframe$Bitcoin))
regime2 <- numeric(length(dframe$Bitcoin))
for (i in 2:length(dframe$Bitcoin)) {if (dframe$Bitcoin[i -
    1] <= threshold) {regime1[i] <- const.L + phiL.1 * dframe$
    Bitcoin[i - 1]} else {regime2[i] <- const.H + phiH.1 *
    dframe$Bitcoin[i - 1] }}
plot(dframe$Date, dframe$Bitcoin, type = "l", col = "black",
    xlab = "Date", ylab = "log of Bitcoin Value in $", main =
    "Fitting of TAR Model and 2 Regime Separation Plot")
lines(dframe$Date, regime1, col = "yellow")
lines(dframe$Date, regime2, col = "green")
legend("top", legend = c("Bitcoin", "Regime 1", "Regime 2"),
     col = c("black", "yellow", "green"), lty = 1, cex = 0.8)
abline(h = threshold, col = "red", lty = 2)
axis.Date(1, at = seq(min(dframe$Date), max(dframe$Date), by
    = "year"))
```

```
Non linear autoregressive model

SETAR model ( 2 regimes)
Coefficients:
Low regime:
   const.L      phiL.1
0.02562248 0.99760588

High regime:
  const.H     phiH.1
0.5247176 0.9523690

Threshold:
-Variable: Z(t) = + (1) X(t)
-Value: 10.84
Proportion of points in low regime: 80.69%       High regime: 19.31%

Residuals:
       Min         1Q     Median         3Q        Max
-0.1773253 -0.0143189 -0.0013176  0.0156176  0.1821717

Fit:
residuals variance = 0.001111,  AIC = -8521, MAPE = 0.2185%

Coefficient(s):

         Estimate  Std. Error  t value Pr(>|t|)
const.L 0.0256225   0.0340481   0.7525  0.45187
phiL.1  0.9976059   0.0032939 302.8638  < 2e-16 ***
const.H 0.5247176   0.2589615   2.0262  0.04295 *
phiH.1  0.9523690   0.0234991  40.5279  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Threshold
Variable: Z(t) = + (1) X(t)

Value: 10.84
```

Figure 3.9: tar model summary 2 regime

The code fits a TAR model.

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

– **Model Fitting:** The TAR model with two regimes was fitted to the data with threshold of 10.471078.

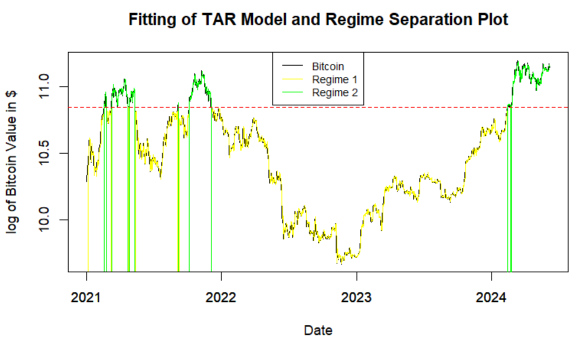


Figure 3.10: Model Fitting of TAR Model

- **Fitted Model observations:**
  - Bitcoin's value shows significant volatility with regime switches in 2021-2022, stabilizing in late 2022-2023, and the TAR model reveals upward or stable trends above the threshold and downward or volatile trends below.

3.3.3 Threshold Autoregressive (TAR) Model for 3 Regimes

- **Model Fitting for 3-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.

```r
library(tsDyn)
tar_model1=setar(dframe$Bitcoin, m = 1, thDelay = 0, nthresh
    = 2, model = "TAR")
summary(tar_model)
```

```
Non linear autoregressive model

SETAR model ( 3 regimes)
Coefficients:
Low regime:
   const.L      phiL.1
0.05674433 0.99444240

Mid regime:
  const.M     phiM.1
0.5953270 0.9437478

High regime:
  const.H     phiH.1
0.1858152 0.9830002

Threshold:
-Variable: Z(t) = + (1) X(t)
-Value: 10.37 10.69
Proportion of points in low regime: 43.34%       Middle regime: 25.46%   High regime: 31.21%

Residuals:
        Min          1Q      Median          3Q         Max
-0.17621164 -0.01449618 -0.00058882  0.01559528  0.18166393

Fit:
residuals variance = 0.001106,  AIC = -8520, MAPE = 0.2182%

Coefficient(s):

         Estimate  Std. Error  t value  Pr(>|t|)
const.L 0.0567443   0.0738657   0.7682  0.442508
phiL.1  0.9944424   0.0073275 135.7144 < 2.2e-16 ***
const.M 0.5953270   0.2254693   2.6404  0.008384 **
phiM.1  0.9437478   0.0213495  44.2047 < 2.2e-16 ***
const.H 0.1858152   0.1263567   1.4706  0.141662
phiH.1  0.9830002   0.0115677  84.9784 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Threshold
Variable: Z(t) = + (1) X(t)

Value: 10.37 10.69
```

Figure 3.11

The above code fits a Threshold Autoregressive (TAR) model with 3 regimes ($m = 3$) to the Bitcoin price series stored in the dataframe `dframe`.

$$
\hat{y}_t = \begin{cases} 0.0567443 + 0.9944424 \cdot y_{t-1}, & \text{if } y_{t-1} \leq 10.37 \\ 0.5953270 + 0.9437478 \cdot y_{t-1}, & \text{if } 10.37 < y_{t-1} \leq 10.69 \\ 0.1858152 + 0.9830002 \cdot y_{t-1}, & \text{if } y_{t-1} > 10.69 \end{cases}
$$

- **Model Fitting:** The TAR model with three regimes was fitted to the data with thresholds of 10.37 and 10.69.

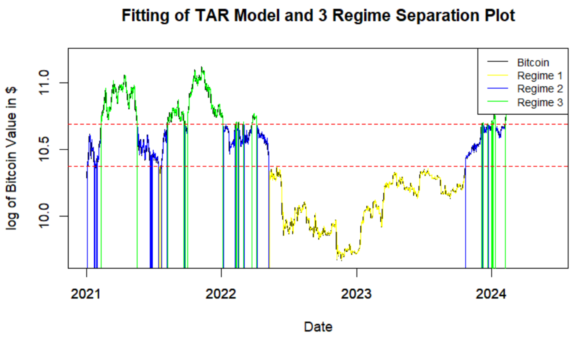


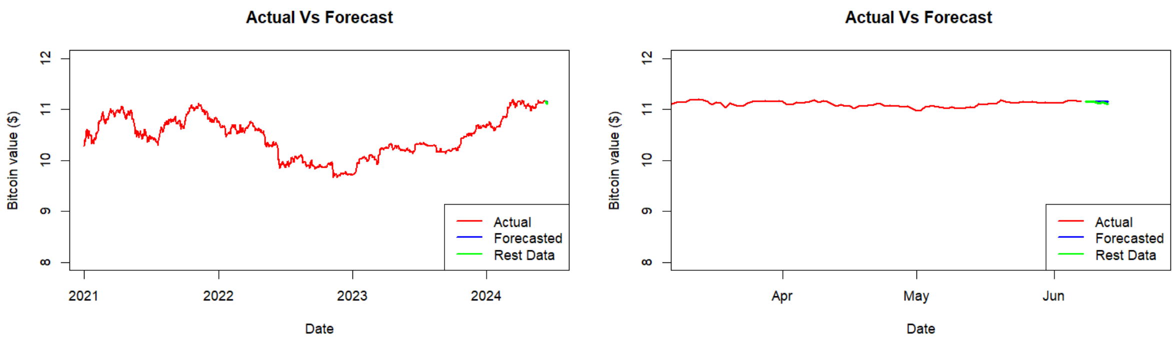
Figure 3.12: Model Fitting of 3-Regime TAR Model

- **Fitted model observations:**
  - The graph(See Figure 3.12) reveals significant volatility in Bitcoin prices, with frequent regime switching, particularly around threshold values.
  - Bitcoin prices remained low and volatile in late 2022 and early 2023, showed high volatility across all regimes in 2024, with high regimes trending upwards, low regimes trending downwards or being volatile, and the mid regime serving as a transitional phase.

3.4 Results

Forecasting using ARIMA model:

We will now forecast the future value using the fitted ARIMA model.



(a) Forecasting Plot (b) Zoomed-in Forecasting Plot

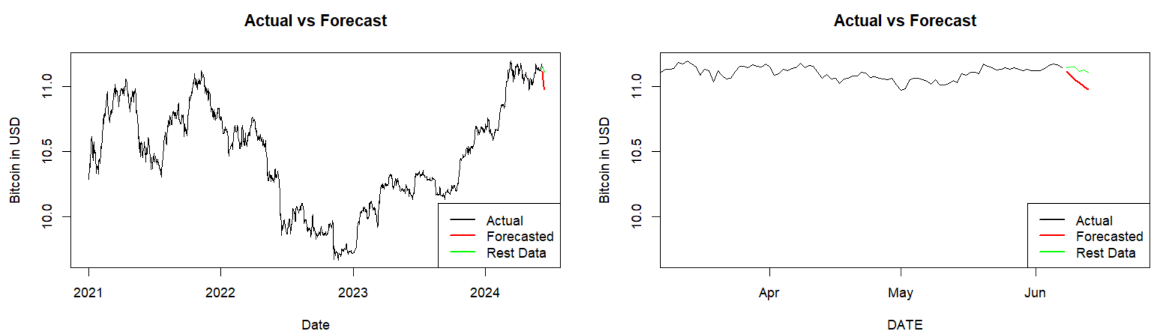
Figure 3.13: Comparison of Forecasting and Zoomed-in Forecasting Plots

Observations from the Plots:

- The ARIMA(1, 1, 0) model effectively captures the overall trend and short-term fluctuations of Bitcoin values,for accuracy comparation we can see zoomed in plot but we can see there is slight deference in actual and forecast plots (See Figure 3.13b).

3.4.1 Forecasting using Tar model for regime 3:

We have seen the forecasting plot using arima model now we will again do forecast with tar model with 3 regime.



(a) Forecasting Plot (b) Zoomed-in Forecasting Plot

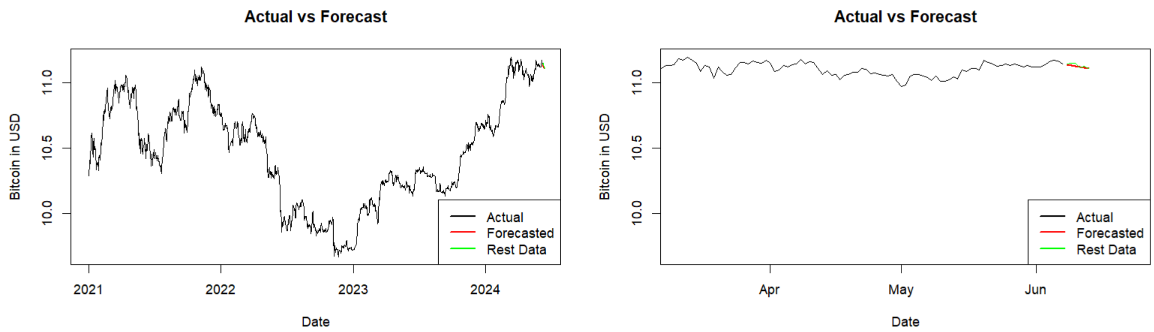
Figure 3.14: Comparison of Forecasting and Zoomed-in Forecasting Plots

Observations from the Plots:

- The TAR model with 3 regimes struggles to capture the overall trend and short-term fluctuations of Bitcoin values, as shown by its lower accuracy in the zoomed-in plot (See Figure 3.14).

3.4.2 Forecasting using Tar model for regime 2:

We have seen ARIMA and TAR 3 Regime model forecasts, now for better accuracy we will forecast using TAR model of 2 ragime.



(a) Forecasting Plot

(b) Zoomed-in Forecasting Plot

Figure 3.15: Comparison of Forecasting and Zoomed-in Forecasting Plots

Observations from the Plots:

- The TAR model with 2 regimes effectively captures the overall trend and short-term fluctuations of Bitcoin values, as shown by the improved accuracy in the zoomed-in plot (See Figure 3.15b).

3.4.3 Model Evaluation

- **Root Mean Square Error (RMSE):**

```
cat("ARIMA model:",rmse(arima_model))
cat("TAR model for 3 regime: ",rmse(tar_model1))
cat("TAR model for 2 regime:",rmse(tar_model))
```

  - ARIMA model: 0.0207
  - TAR model for 3 regime: 0.09506
  - TAR model for 2 regime: 0.0129

- **Akaike Information Criterion (AIC):**

```
cat("ARIMA model:",aic(arima_model)
cat("TAR model for 3 regime: ",aic(tar_model1)
cat("TAR model for 2 regime:",aic(tar_model))
```

  - ARIMA model: -4959.16
  - TAR model for 3 regime: -8519.58
  - TAR model for 2 regime: -8520.53

- **Model Performance:**

Model	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6
ARIMA Forecast	11.14674	11.14671	11.14671	11.14671	11.14671	11.14671
TAR (2 Regime) Forecast	11.13979	11.13391	11.12830	11.12297	11.11789	11.11305
TAR (3 Regime) Forecast	11.11431	11.08443	11.05623	11.02962	11.00451	10.98081
Actual Values	11.14629	11.15131	11.14895	11.11716	11.13124	11.10892

Table 3.3: Forecasted and Actual Values for the Next 6 Days

3.5 Conclusion of the Case Study Project

This case study focused on analyzing and forecasting the daily closing prices of Bitcoin in U.S. Dollars from December 1, 2014, to June 23, 2024, using various time series models, including ARIMA and Threshold Autoregressive (TAR) models with 2 and 3 regimes.

Key findings from the analysis include:

- **Model Performance and Accuracy:** The 2-regime TAR model outperformed the ARIMA and 3-regime TAR models with the lowest RMSE (0.0129) and best AIC (-8520.53), providing the most accurate and parsimonious predictions.
- **Forecasting and Dynamics:** The 2-regime TAR model offered the most robust short-term forecasts closely matching actual data, while TAR models effectively captured Bitcoin's nonlinear dynamics and volatility, unlike the ARIMA model.
- **Model Selection:** The study emphasized the effectiveness of simpler models like the 2-regime TAR for financial time series, highlighting their superior accuracy and reliability compared to more complex models.

In conclusion, the 2-regime TAR model emerged as the most effective for forecasting Bitcoin prices, balancing accuracy and simplicity, and underscoring the importance of nonlinear modeling techniques for financial time series data.

Chapter 4

Study Material Project

I was involved in the creation of study material for a workshop conducted at IIT Bombay on ”Linear Time Series”. The study material created was on time series linear models, including AutoRegressive (AR), Moving Average (MA), and AutoRegressive Moving Average (ARMA) models. Two documents were created. The first document focused on the coding aspect, i.e., the practical usage of the models, and the second one covered the underlying theoretical concepts of the code. The study aims to build and demonstrate the foundation required for the practical applications of these models in time series data forecasting. The study material included simulation of the data, model fitting, residual analysis, ACF and PACF plots, and forecasting results using each model. This resource can be accessed using the following link:

Link to Study Material: `https://r.fossee.in/resources`

Chapter 5

Conclusion

During the FOSSEE Summer Fellowship, I advanced my knowledge in the use of R language through several projects. The Textbook Companion (TBC) project involved coding solved problems from a standard textbook in R. The case study done on Bitcoin data provided valuable insights into time series analysis and cryptocurrency price forecasting. Additionally, the Study Material Project contributed to enhancing my knowledge in time series linear models and can serve as a valuable resource to people interested in time series analysis.

My FOSSEE fellowship experience was enriching and impactful. As this was done in campus, it facilitated collaborative learning and interaction with peers and mentors. It equipped me with valuable skills and methodologies that I can leverage in future endeavors. Overall, this fellowship not only enhanced my technical proficiency but also provided a deeper understanding of organizational dynamics and societal contributions.

Bibliography

[1] Coinbase. Coinbase, 2024. `https://www.coinbase.com/`.

[2] Coinbase. User agreement - united states, 2024. Accessed: 2024-07-22.

[3] Federal Reserve Bank of St. Louis. Bitcoin data series, 2024. `https://fred.stlouisfed.org/series/CBBTCUSD`.

[4] John E. Freund, Irwin Miller, and Marylees Miller. *Mathematical Statistics with Applications*. Pearson Education, 2014.

[5] R.J. Hyndman and Y. Khandakar. Automatic time series for forecasting: The forecast package for r. Technical report, Monash University, Department of Econometrics and Business Statistics, 2007.

[6] S. J. McKean, S. E. S. (Stephen), and P. L. (Paul). *Nonlinear Time Series: Theory, Methods, and Applications with R Examples*. Springer, 2020.

[7] Robert H. Shumway and David S. Stoffer. *Time Series Analysis and Its Applications with R Examples*. Springer, 2020.

[8] Ruey S. Tsay. *Analysis of Financial Time Series*. Wiley, 2010.

[9] Ruey S. Tsay and Rong Chen. *Nonlinear Time Series Analysis*. Wiley, 2019.

[10] I. M. Wirawan, T. Widiyaningtyas, and M. M. Hasan. Short term prediction on bitcoin price using arima method. In *Proceedings of the 2019 International Seminar on Application for Technology of Information and Communication (iSemantic)*, pages 260–265. IEEE, 2019.