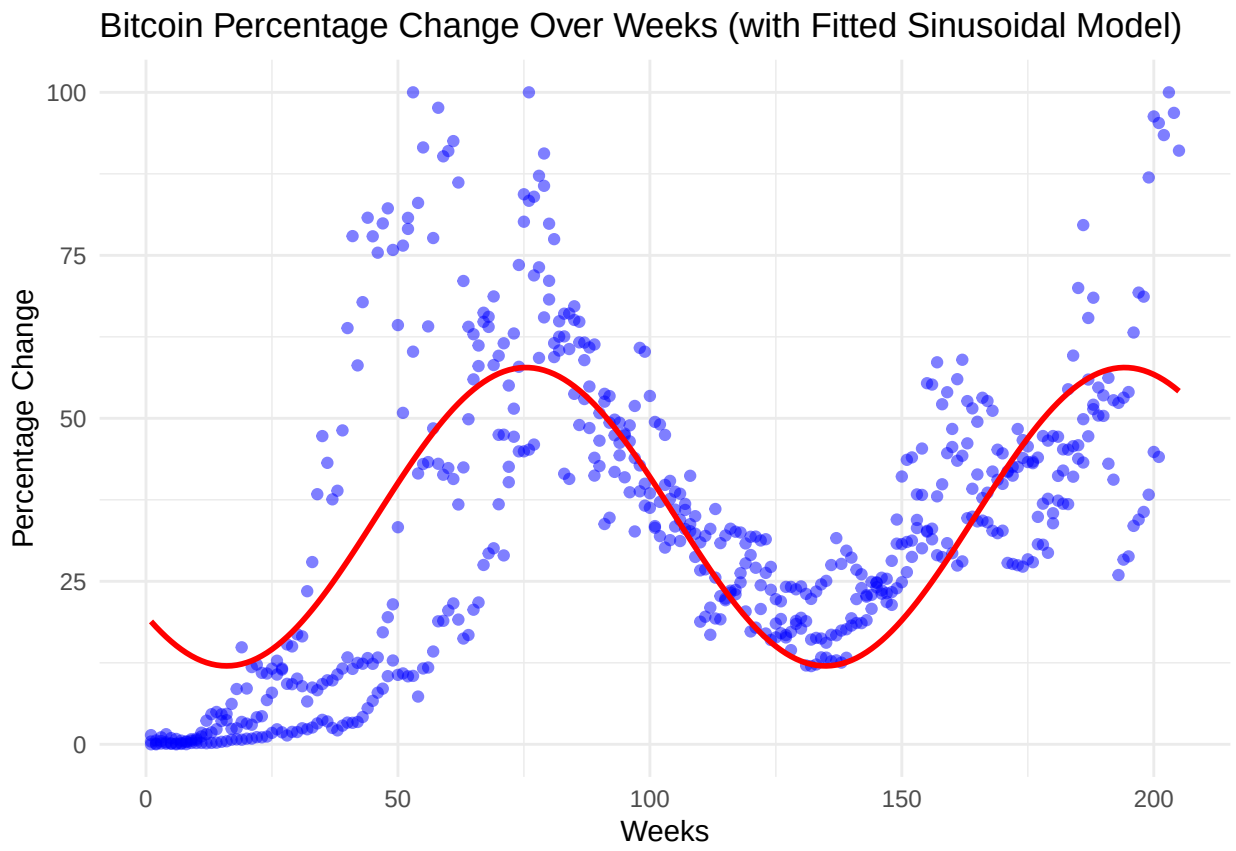


The graph shows the distribution of the normalized values and how closely they adhere to the predicted sinusoidal cycles. The fit of Bitcoin's normalized price closely follows the prediction, yet still has notable variability.

Both tests indicate existence of causal cyclical patterns in Bitcoin's price movement. This is confirmed by the significance of the p-values. However, the residual errors are also considerable too, at 11.41 and 16.06. These indicates that price value and movement variability isn't fully explained by cyclicality. This is as expected as no equity's or currency's price can be entirely predicted. Prices are often influenced by random occurences and variables in the market including inflation, news, stock market performance, political events, and investor sentiment. The key takeaway is that Bitcoin's price movement does exhibit a statistically significant cyclical pattern.

Forecasting Performance

Calculating Halving Period Percentage Change.

This code section is needed for forecasting a percentage return for the current halving period (2024-2028). The start and finish values of each halving period are needed to show how much Bitcoin may have grown. The resulting dataframe shows that the greatest growth occurred in 2012 to 2016 at over 5000%. The more recent 2020 to 2024 is the lowest at only 650% gain.

```
# Function to filter daily data, with error catch
filter_halving_period_daily <- function(start_date, end_date) {
  start_date <- as.Date(start_date)
```

```r
  end_date <- as.Date(end_date)

  filtered_data <- df %>%
    filter(timestamp >= start_date & timestamp < end_date)

  if (nrow(filtered_data) == 0) {
    warning("No data found for this period.")
  }

  return(filtered_data)
}

# Apply the filter function the daily data
halving_2012_to_2016 <- filter_halving_period_daily("2012-11-28", "2016-07-09")
halving_2016_to_2020 <- filter_halving_period_daily("2016-07-09", "2020-05-11")
halving_2020_to_2024 <- filter_halving_period_daily("2020-05-11", "2024-04-19")


# Function to calculate the metrics for each halving period, including last price of value the halving

calculate_halving_metrics <- function(data) {
  start_value <- data$close[1]
  finish_value <- data$close[nrow(data)]
  percentage_change_gain <- ((finish_value - start_value) / start_value) * 100

  return(data.frame(
    start_value = start_value,
    finish_value = finish_value,
    percentage_change_gain = percentage_change_gain
  ))
}

# Apply the function to each halving period daily data
halving_2012_to_2016_metrics <- calculate_halving_metrics(halving_2012_to_2016)
halving_2016_to_2020_metrics <- calculate_halving_metrics(halving_2016_to_2020)
halving_2020_to_2024_metrics <- calculate_halving_metrics(halving_2020_to_2024)

# Combine the metrics for all periods into one table
halving_summary_table <- rbind(
  cbind(period = "2012 to 2016", halving_2012_to_2016_metrics),
  cbind(period = "2016 to 2020", halving_2016_to_2020_metrics),
  cbind(period = "2020 to 2024", halving_2020_to_2024_metrics)
)

print(halving_summary_table)
```

```
##         period start_value finish_value percentage_change_gain
## 1 2012 to 2016    12.37713      666.523              5285.1157
## 2 2016 to 2020   650.96002     8756.431              1245.1565
## 3 2020 to 2024  8601.79620    63512.753               638.3662
```

Growth Rate and Forecasting

This section is calculating the growth rates for each halving periods. However, the rate is not decreasing linearly. Using exponential interpolation of price growth between the last three periods, we can create a more accurate forecast of the growth rate for the current period, 2024 to 2028. This comes out to 96.57%.

Additionally, using the exponential model's residuals and percentage differences b/w predicted and actual, the marginal errors rates can be calculated. This will shows how te range of value this forecast could predicts. The percentage error rate is 18%, ranging from 78% to 114% growth. Residuals' error rate is 144%, ranging from -48% loss to 241% growth.

```
# Extract relevant table data
percentage_change_data <- halving_summary_table[, c("period", "percentage_change_gain")]

# Convert the period to a factor for correct ordering
percentage_change_data$period <- factor(percentage_change_data$period, levels = c("2012 to 2016", "2016

# Create an index for periods for ordering
percentage_change_data$period_numeric <- as.numeric(percentage_change_data$period)

# Create the exponential model, intial paramater guess a = 500, b = -.5
exp_model <- nls(percentage_change_gain ~ a * exp(b * period_numeric),
                 data = percentage_change_data,
                 start = list(a = 500, b = -0.5))

# Fit the model to the percentage gain
predicted_values <- predict(exp_model, newdata = data.frame(period_numeric = 1:3))

# Add predicted values back to the original percentage_change_data dataframe
percentage_change_data$predicted_value <- predicted_values
percentage_change_data$residuals <-  percentage_change_data$percentage_change_gain - predicted_values
percentage_change_data$percentage_difference <- abs(percentage_change_data$residuals) /
                                                 percentage_change_data$percentage_change_gain * 100

# Calculating the marginal error
marginal_error_percentage <- mean(percentage_change_data$percentage_difference)

# Using model, predict the percentage gain for the halving period 2024 to 2028
predicted_2024_to_2028_exp <- predict(exp_model, newdata = data.frame(period_numeric = 4))

# Store the predicted data as an observation, NA input for the lack of residual and percentage change d
new_period_data_exp <- data.frame(
  period = "2024 to 2028",
  percentage_change_gain = predicted_2024_to_2028_exp,
  predicted_value = predicted_2024_to_2028_exp,
  period_numeric = 4,
  residuals = NA,
  percentage_difference = NA
)

# Store the predicted observation with the orginal dataframe
percentage_change_data_with_exp_prediction <- rbind(percentage_change_data, new_period_data_exp)

# Add the halving period 2024-2028 to the period set
percentage_change_data_with_exp_prediction$period <- factor(
```

```
  percentage_change_data_with_exp_prediction$period,
  levels = c("2012 to 2016", "2016 to 2020", "2020 to 2024", "2024 to 2028")
)

print(percentage_change_data_with_exp_prediction)
```

```
##         period percentage_change_gain period_numeric predicted_value  residuals
## 1 2012 to 2016              5285.1157              1       5266.1897   18.92602
## 2 2016 to 2020              1245.1565              2       1388.6976 -143.54111
## 3 2020 to 2024               638.3662              3        366.2005  272.16571
## 4 2024 to 2028                96.5673              4         96.5673         NA
##   percentage_difference
## 1             0.3581003
## 2            11.5279575
## 3            42.6347327
## 4                    NA
```

```
# Calculate the lower and upper bounds using the residuals and marginal error
residual_bound <- mean(abs(percentage_change_data$residuals))
lower_bound_residual <- predicted_2024_to_2028_exp - residual_bound
upper_bound_residual <- predicted_2024_to_2028_exp + residual_bound
lower_bound_percentage <- predicted_2024_to_2028_exp - marginal_error_percentage
upper_bound_percentage <- predicted_2024_to_2028_exp + marginal_error_percentage


s <- paste("Predicted % Change for 2024 to 2028:", predicted_2024_to_2028_exp,"\n\n", "Residual Bound (a
"Lower Bound (Residual-based) for 2024 to 2028:", lower_bound_residual,"\n",
"Upper Bound (Residual-based) for 2024 to 2028:", upper_bound_residual,"\n\n",
"Marginal Error Percentage:", marginal_error_percentage,"\n",
"Lower Bound (%-based) for 2024 to 2028:", lower_bound_percentage,"\n",
"Upper Bound (%-based) for 2024 to 2028:", upper_bound_percentage)

cat(s)
```

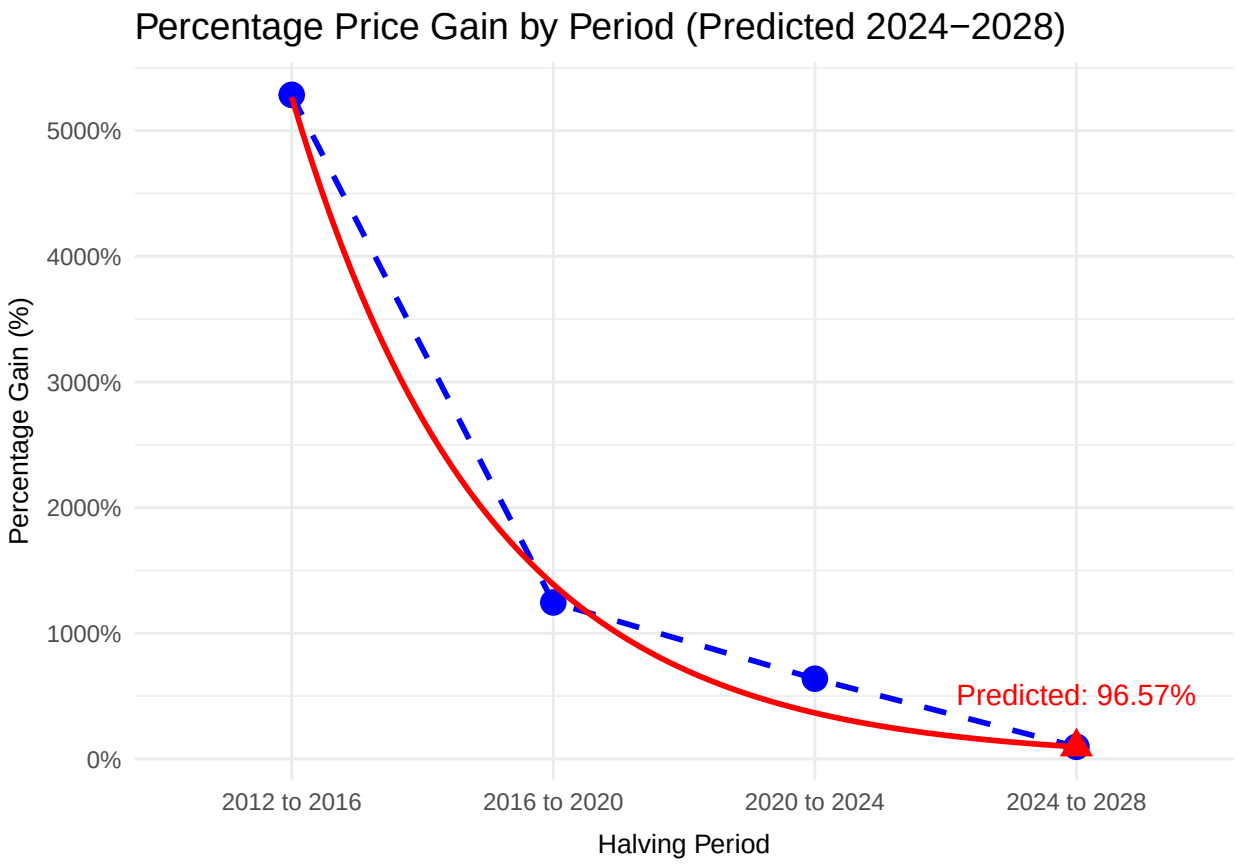
```
## Predicted % Change for 2024 to 2028: 96.5672983264403
##
##  Residual Bound (average absolute residuals): 144.877614487125
##  Lower Bound (Residual-based) for 2024 to 2028: -48.3103161606843
##  Upper Bound (Residual-based) for 2024 to 2028: 241.444912813565
##
##  Marginal Error Percentage: 18.1735968679651
##  Lower Bound (%-based) for 2024 to 2028: 78.3937014584753
##  Upper Bound (%-based) for 2024 to 2028: 114.740895194405
```

Exponential Forecast

Graphing the price growth of the last three halving periods, we can see a decreasing growth rate. Using an exponential growth rate shows a close fitting model.

```
#Plot the expontential forecast and the actual percentage gain
p <- ggplot(percentage_change_data_with_exp_prediction,
       aes(
```

```
        x = period,
        y = percentage_change_gain,
        group = 1)
      ) +
  geom_point(
    size = 4,
    color = "blue") +

  geom_line(
    color = "blue",
    linetype = "dashed",
    linewidth = 1) +

  labs(
    title = "Percentage Price Gain by Period (Predicted 2024-2028)",
    x = "Halving Period",
    y = "Percentage Gain (%)") +
  theme_minimal() +

  geom_smooth(
    method = "nls",
    formula = y ~ a * exp(b * x),
    method.args = list(start = list(a = 500, b = -0.5)),
    se = FALSE,
    color = "red",
    linetype = "solid") +

  theme(
    plot.title = element_text(hjust = 0, size = 14),
    axis.title.x = element_text(vjust = -1, size = 10),
    axis.title.y = element_text(vjust = 2, size = 10)
  ) +
  scale_y_continuous(
    labels=function(x)
    paste0(x,"%")) +

  geom_point(
    data = new_period_data_exp,
    aes(
      x = period,
      y = percentage_change_gain),
    color = "red",
    size = 4,
    shape = 17) +

  annotate("text",
           x = 4,
           y = predicted_2024_to_2028_exp,
           label = paste0("Predicted: ",
                          round(predicted_2024_to_2028_exp, 2), "%"),
           vjust = -2, color = "red", size = 4)

print(p)
```



Price Forecast

This analysis shows the price Bitcoin is expected to rise to if the 96.57% growth rate is accurate. Bitcoin's starting price in the current halving period is approximately $65,000 but could grow to $127,000 by 2028.

```
# Filter data for only post-2024, April 19th, Bitcoin halving
post_2024_weekly_data <- weekly_data %>%
  filter(timestamp >= as.Date("2024-04-19"))

# Create an index for post-2024 data using weeks as index again
post_2024_weekly_data$index <- seq_len(nrow(post_2024_weekly_data))

# Extracting the starting close price
start_value_2024 <- post_2024_weekly_data$close[1]

# Calculate the predicted final value for 2028 by multiplying the starting value by the predicted perce
predicted_final_value_2028 <- start_value_2024 * (1 + predicted_2024_to_2028_exp/100)
predicted_monetary_diff_2024_to_2028 <- predicted_final_value_2028 - start_value_2024

s <- paste("Prices of Bitcoin at Start and End of 2024 to 2028 Halving Period\n",
    "START: ", dollar(start_value_2024), "\n",
    "END: \t", dollar(predicted_final_value_2028))

cat(s)
```

```
## Prices of Bitcoin at Start and End of 2024 to 2028 Halving Period
##  START:  $64,926.64
##  END:     $127,625
```

Error Value: Upper & Lower Bounds

This section shows the range of predicted values of Bitcoin by 2028. Prediction is not perfect and there's highly likeliness that Bitcoin won't adhere to the 96.57% gain prediction. The analysis below shows the ranges of upper and lower bounds calculated by the error values from the exponential forecast. The percentage range, is more compact, with values ranging from $115,000 to $139,000. This shows growth from the starting value of nearly $65,000. Residuals, however, show a much larger range from $33,000 to $221,000. This includes 50%+ loss . The range of the variability is much higher for residuals.

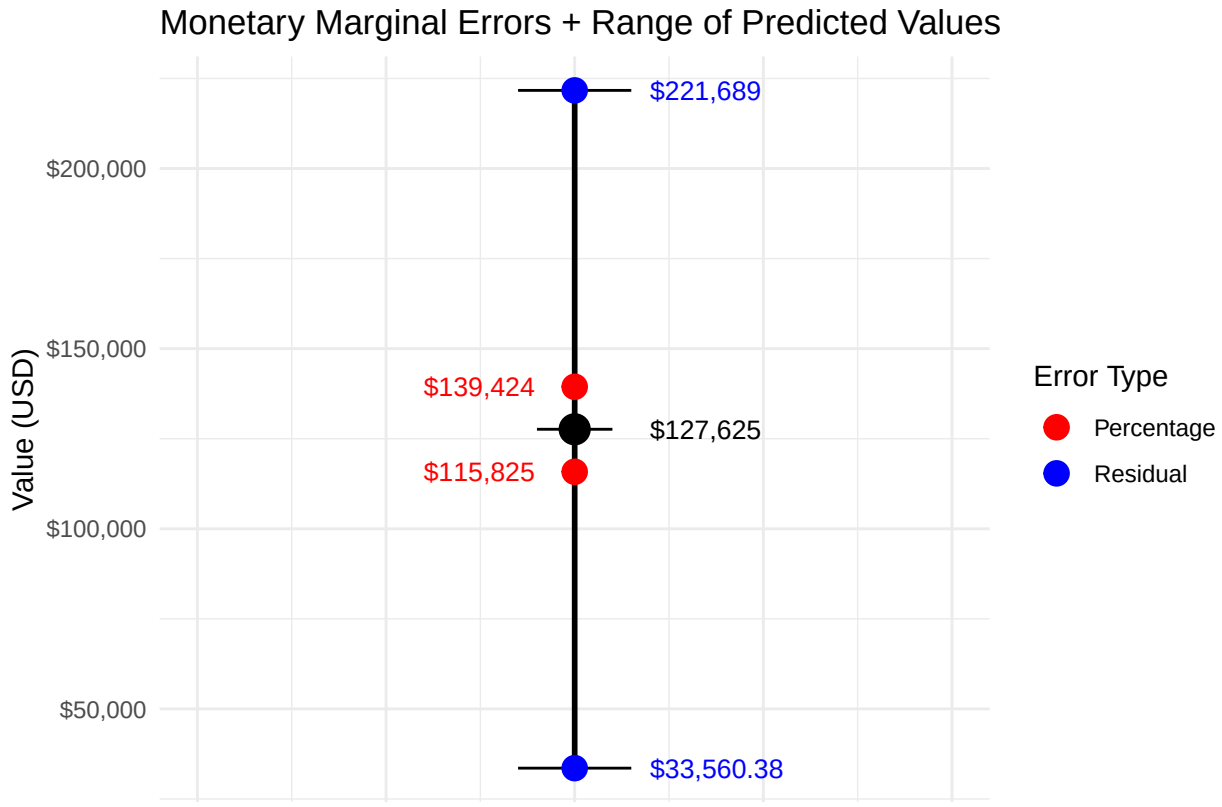
```
# Calculate the upper and lower bounds, given by multiplying the error/residual percentages by the doll
lower_bound_residual_monetary <- (1 + lower_bound_residual / 100) * start_value_2024
upper_bound_residual_monetary <- (1 + upper_bound_residual / 100) * start_value_2024

lower_bound_percentage_monetary <- (1 + lower_bound_percentage / 100) * start_value_2024
upper_bound_percentage_monetary <- (1 + upper_bound_percentage / 100) * start_value_2024

#Create a table with error monetary values, with labels for graphs
error_monetary_data <- data.frame(
  value = c(lower_bound_residual_monetary, upper_bound_residual_monetary,
            lower_bound_percentage_monetary, upper_bound_percentage_monetary,
            predicted_final_value_2028),
  error_type = c("Residual", "Residual", "Percentage", "Percentage", "Predicted")
)

# Plot range of values for the END of 2024 - 2028 halving periods
p <- ggplot() +

  # Vertical line to indicate the range of value plausible by the prediction model error values
  geom_segment(
    aes(
      x = 1,
      xend = 1,
      y = min(
        c(
          lower_bound_residual_monetary,
          lower_bound_percentage_monetary)),
          yend = max(
            c(
              upper_bound_residual_monetary,
              upper_bound_percentage_monetary))),
    color = "black",
    linewidth = 1) +

  #Line Segments to denoted the midpoint and the endpoints of the range
  geom_segment(
    aes(x = 0.985, xend = 1.015,
        y = lower_bound_residual_monetary),
    color = "black", size = 0.5) +
  geom_segment(
```

```
  aes(x = 0.985, xend = 1.015,
      y = upper_bound_residual_monetary),
  color = "black", size = 0.5) +
geom_segment(
  aes(x = 0.99, xend = 1.01,
      y = predicted_final_value_2028),
  color = "black", linewidth = 0.5) +

# Data Points of the monetary values for residuals and percentage errors, and the midpoint in black
geom_point(data = error_monetary_data[1:2, ],
           aes(x = 1, y = value,
               color = "Residual"),
           size = 4) +
geom_point(data = error_monetary_data[3:4, ],
           aes(x = 1, y = value,
               color = "Percentage"),
           size = 4) +
geom_point(
  aes(x = 1,
      y = predicted_final_value_2028),
  size = 5,
  color = "black") +

# Dollar Labels for the points, Red for Percentage, Blue for Residual
geom_text(
  aes(x = 1.02,
      y = lower_bound_residual_monetary,
      label = scales::dollar(lower_bound_residual_monetary)),
  hjust = 0, size = 3.5, color = "blue") +
geom_text(
  aes(x = 1.02,
      y = upper_bound_residual_monetary,
      label = scales::dollar(upper_bound_residual_monetary)),
  hjust = 0, size = 3.5, color = "blue") +
geom_text(
  aes(x = 0.96,
      y = lower_bound_percentage_monetary,
      label = scales::dollar(lower_bound_percentage_monetary)),
  hjust = 0, size = 3.5, color = "red") +
geom_text(aes(x = 0.96,
              y = upper_bound_percentage_monetary,
              label = scales::dollar(upper_bound_percentage_monetary)),
          hjust = 0, size = 3.5, color = "red") +
geom_text(
  aes(x = 1.02,
      y = predicted_final_value_2028,
      label = scales::dollar(predicted_final_value_2028)),
  hjust = 0, size = 3.5, color = "black") +

#
labs(title = "Monetary Marginal Errors + Range of Predicted Values",
     x = "",
     y = "Value (USD)",
```

```
      color = "Error Type") +
  theme_minimal() +
  coord_cartesian(xlim = c(0.9, 1.1)) +
  theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank()) +
  scale_y_continuous(labels = dollar_format()) +
  scale_color_manual(values = c("Residual" = "blue", "Percentage" = "red"))
```

```
## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.
```

```
print(p)
```

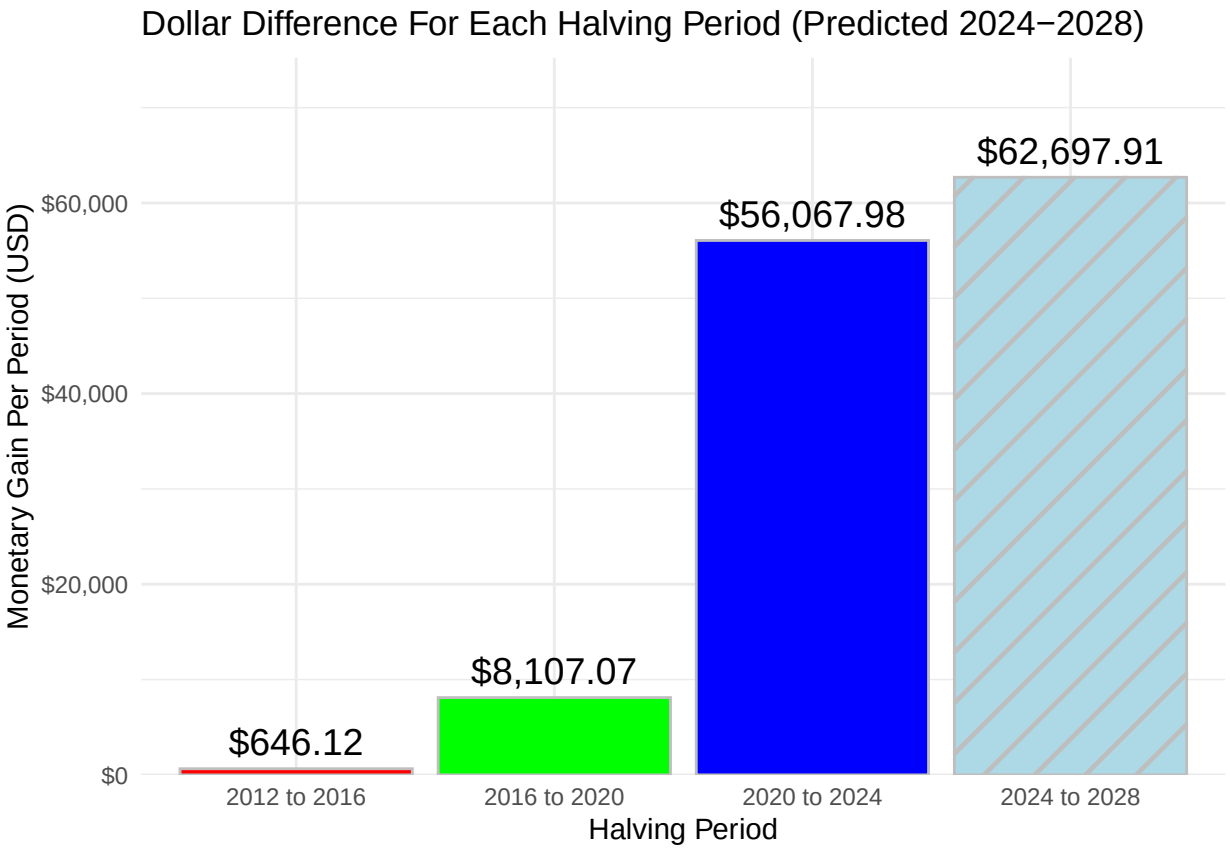


Price Gain Over Halving Periods

This analysis below shows the pattern of price gain of Bitcoin over the last 3 halving periods and the predicted price gain by 2028 if the predicted value 96.57% gain holds. Again, if this prediction is accurate, it shows slowing growth from the previous periods, which have had massive jumps in value. However, even though the percentage value is smaller, the predicted value will still increase by over $62,000 in price.

```r
# Define the start and final values for each halving period
start_value_2012_to_2016 <- halving_2012_to_2016_weekly$close[1]
final_value_2012_to_2016 <- halving_2012_to_2016_weekly$close[nrow(halving_2012_to_2016_weekly)]

start_value_2016_to_2020 <- halving_2016_to_2020_weekly$close[1]
final_value_2016_to_2020 <- halving_2016_to_2020_weekly$close[nrow(halving_2016_to_2020_weekly)]

start_value_2020_to_2024 <- halving_2020_to_2024_weekly$close[1]
final_value_2020_to_2024 <- halving_2020_to_2024_weekly$close[nrow(halving_2020_to_2024_weekly)]

# Take the difference between final and close prices
monetary_diff_2012_to_2016 <- final_value_2012_to_2016 - start_value_2012_to_2016
monetary_diff_2016_to_2020 <- final_value_2016_to_2020 - start_value_2016_to_2020
monetary_diff_2020_to_2024 <- final_value_2020_to_2024 - start_value_2020_to_2024


# Data for the bar chart of monetary differences
bar_data <- data.frame(
  period = c("2012 to 2016", "2016 to 2020", "2020 to 2024"),
  monetary_difference = c(
    monetary_diff_2012_to_2016,
    monetary_diff_2016_to_2020,
    monetary_diff_2020_to_2024)
)

# Create and add the observation for the 2024-2028 halving period
predicted_data <- data.frame(
  period = "2024 to 2028",
  monetary_difference = predicted_monetary_diff_2024_to_2028,
  pattern = "slanted_lines"
)

bar_data_combined <- rbind(
  data.frame(bar_data, pattern = "solid"),
  predicted_data
)

# Plot the bar graph with the predicted value distinguished by thin slanted lines
p <- ggplot(
  bar_data_combined,
  aes(
    x = period,
    y = monetary_difference,
    fill = period,
    pattern = pattern)) +

  # Stylize the bar data, using unique style to denote the predicted value from the actuals
  geom_bar_pattern(
    stat = "identity",
    aes(
      pattern_fill = pattern),
    color = "grey",
    size = 0.5,
```

```
    pattern_color = "grey",
    pattern_density = 0.05,
    pattern_angle = 45,
    pattern_spacing = 0.05) +
  scale_pattern_manual(
    values = c(
      "solid" = NA,
      "slanted_lines" = "stripe")
    ) +
  scale_fill_manual(
    values = c("red", "green", "blue", "lightblue")) +

  # Add labels and format the y-axis to be dollar values
  scale_y_continuous(
    labels = dollar_format(prefix = "$", suffix = "", big.mark = ","),
                      expand = c(0, 0),
                      limits = c(0,
                         max(bar_data_combined$monetary_difference) * 1.2)) +
  geom_text(
    aes(
      label = scales::dollar(monetary_difference)
      ),
    vjust = -0.5,
    size = 5,
    color = "black") +
  labs(
    title = "Dollar Difference For Each Halving Period (Predicted 2024-2028)",
       x = "Halving Period",
       y = "Monetary Gain Per Period (USD)") +
  theme_minimal() +
  theme(legend.position = "none")

print(p)
```



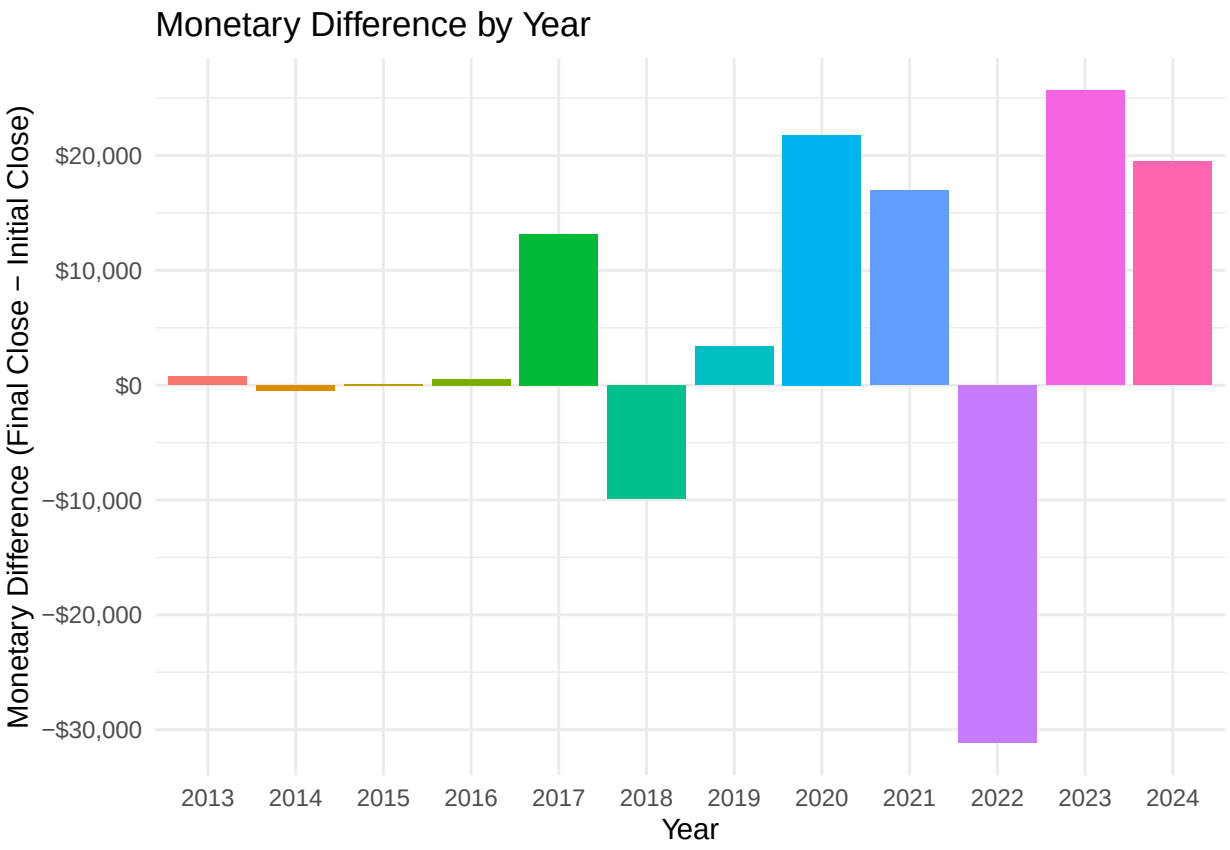
Monetary Performance

Yearly Gain/Loss

In this section, the monetary performance of Bitcoin is being assessed by looking at yearly gain or loss. This will help to answer the question, can this cycle been seen in yearly price performance. Yearly gain/loss is calculated by taking the difference of prices at the beginning and end of the year.

```
# Function to calculate the monetary difference by year

calculate_monetary_diff_by_year <- function(data) {
  data$year <- format(data$timestamp, "%Y")
  monetary_diff_yearly <-
    data %>%
    group_by(year) %>%
    summarize(monetary_difference = last(close) - first(close))

  return(monetary_diff_yearly)
}


# Apply the function to that prices of the dataframe
monetary_diff_yearly <- calculate_monetary_diff_by_year(df)

# Exlcude years prior 2013 due to lack of data
```

```r
monetary_diff_yearly <- monetary_diff_yearly %>%
  filter(year %in% c("2013", "2014", "2015", "2016",
                     "2017", "2018", "2019", "2020",
                     "2021", "2022", "2023", "2024"))

# Bar graph to show the monetary gain/loss each year
p <- ggplot(
  monetary_diff_yearly,
  aes(
    x = year,
    y = monetary_difference,
    fill = year)
  ) +
  geom_bar(stat = "identity") +
  labs(
    title = "Monetary Difference by Year",
    x = "Year",
    y = "Monetary Difference (Final Close - Initial Close)"
    ) +
  theme_minimal() +
  theme(legend.position = "none") +
  scale_y_continuous(labels = dollar_format())

print(p)
```



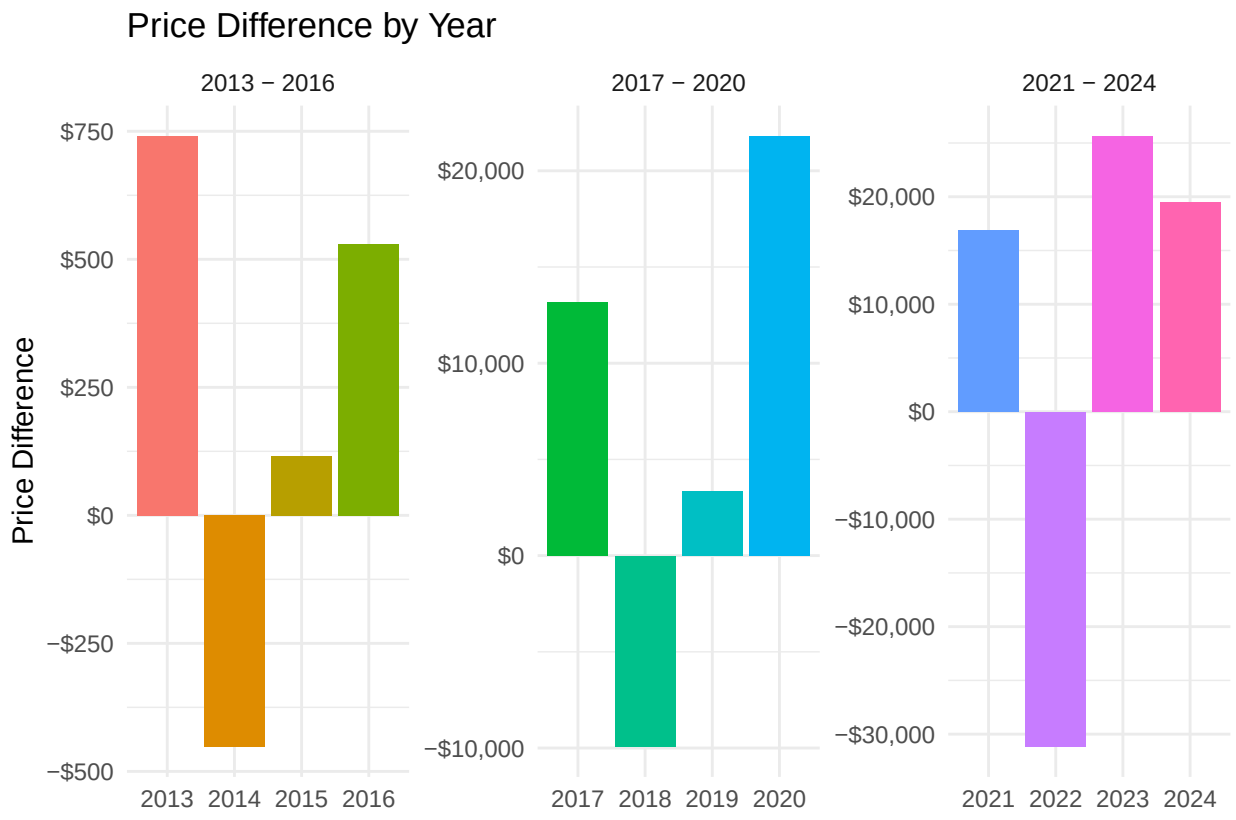
First, observing the data, it's difficult to ascertain price performance. Again, this is due to the Bitcoin's

growth over the past 10+ years, the larger current prices make it difficult to see patterns.

Scaled Yearly Gain/Loss

Following similar steps conducted earlier in analyzing the halving periods, the cycle pattern can be seen more clearly. This involves making sure each 4 year grouping is on a separate y-axis. This ensures a fairness of scale when observing the data.

```
# Organize years into 4 year periods again, for scalability
monetary_diff_yearly <- monetary_diff_yearly %>%
  mutate(period_group = case_when(
    year %in% c("2013", "2014", "2015", "2016") ~ "2013 - 2016",
    year %in% c("2017", "2018", "2019", "2020") ~ "2017 - 2020",
    year %in% c("2021", "2022", "2023", "2024") ~ "2021 - 2024"
  ))

# Plot the bar graph for each 4-year period using facet_wrap for visual scalability
p <- ggplot(
  monetary_diff_yearly,
  aes(
    x = year,
    y = monetary_difference,
    fill = year)
  ) +
  geom_bar(stat = "identity") +
  labs(
    title = "Price Difference by Year",
    x = "",
    y = "Price Difference"
    ) +
  facet_wrap(~ period_group, scales = "free", ncol = 3) +
  theme_minimal() +
  scale_y_continuous(
    labels = scales::dollar_format(prefix = "$", big.mark = ",")
    ) +
  theme(legend.position = "none")

print(p)
```



As shown above, once the data is on a more equal scaling, a clear cycle pattern emerges. The first year shows growth, followed by a price collapse in the second year, slow regain in the third year. Finally major growth spike in the fourth year. Note, that 2024 doesn't show this spike. This may be due to the fact at the time of writing, 2024 is not over yet and maybe missing data.

The overall conclusion of this section is to show even though growth percentage is slowing, the monetary growth is still high and follows a similar cycle.

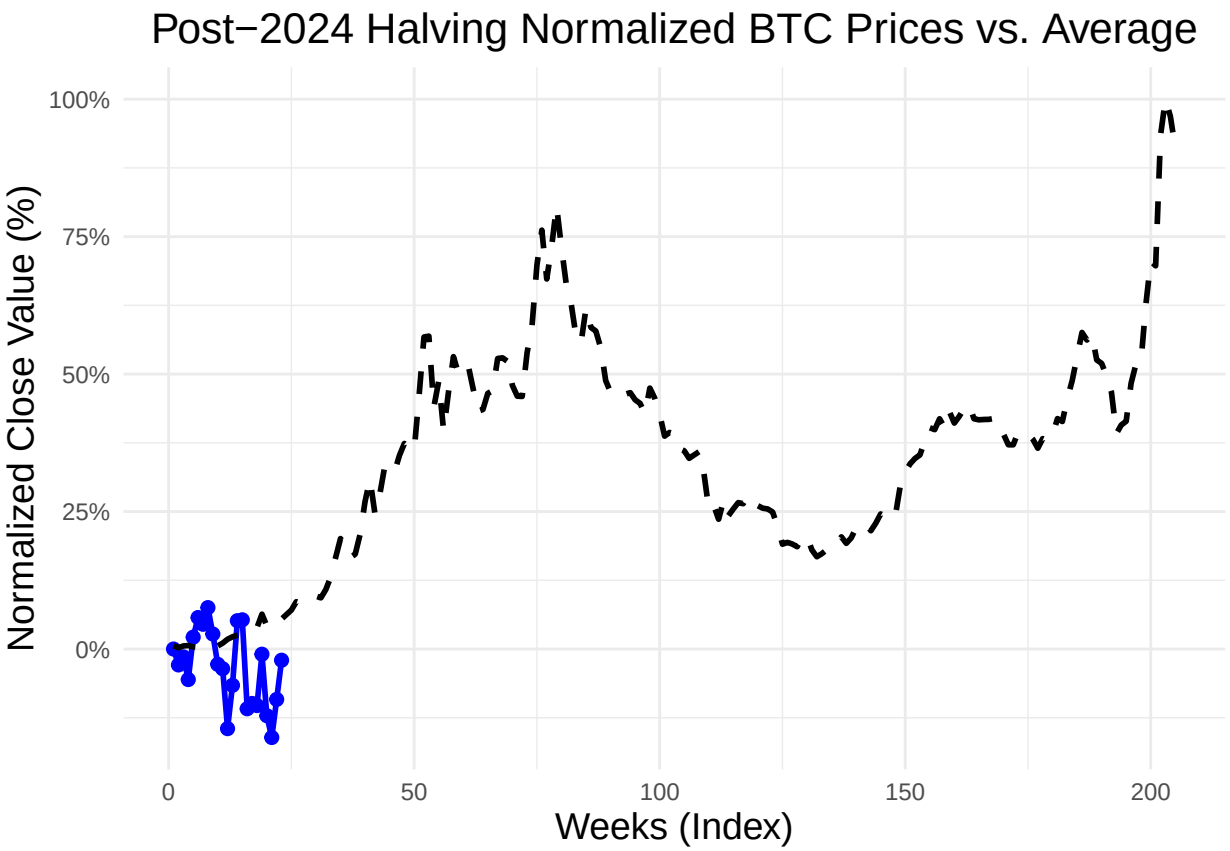
Current Cycle

Post-April 2024 Halving Period

Given the cyclical patterns we have seen, where does this leave the current period? The new halving period started April 19th, 2024. As seen below, we can compared to the average cycle we calculated before. This shows us just how far into the current cycle we are.

```
# Normalize the post-2024 close values based on the predicted final value
post_2024_weekly_data$close_normalized <- (post_2024_weekly_data$close - start_value_2024) / (predicted_

# Plot the normalized post-2024 data against the average line (dashed black line)
p <- ggplot() +
  geom_line(
    data = post_2024_weekly_data,
    aes(
```

```
      x = index,
      y = close_normalized),
    color = "blue",
    linewidth = 1) +
  geom_point(
    data = post_2024_weekly_data,
    aes(
      x = index,
      y = close_normalized),
    color = "blue",
    size = 2) +

  geom_line(
    data = average_normalized_data,
    aes(
      x = index,
      y = average_close_normalized),
    color = "black",
    linetype = "dashed",
    linewidth = 1) +

  labs(
    title = "Post-2024 Halving Normalized BTC Prices vs. Average",
    x = "Weeks (Index)",
    y = "Normalized Close Value (%)") +

  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 16),
    axis.title.x = element_text(size = 14),
    axis.title.y = element_text(size = 14)
  ) +
  scale_y_continuous(labels = scales::label_percent(scale = 1))

print(p)
```



Conclusion

****Disclaimer:**** I am not a professional investment advisor, and this material does not constitute investment advice. Please conduct your own research and due diligence before making any investment decisions.

In this analysis, we have seen how Bitcoin follows a cyclical pattern after each halving. This pattern follows nearly the same rises and falls in price behavior over the same 4 year time spans. It's statistically significant and could produce a return over $62,000 if the cycle continues. The future is never clear and using past performance is not a guaranteed indicator of future returns. Overall, this offers a tempting opportunity given the cyclical pattern.

Patterns like these in market movements could offer lucrative opportunities for investors. Knowing that a particular equity's or currency's future performance allows investors to invest confidently knowing that their money is safe. With these results, it's feasible that Bitcoin may be such an investment.

Now, with the advent of crypto-currency backed ETFs, investors no longer have to directly buy Bitcoin. New strategies for investing in Bitcoin can take place by investing indirectly through these ETFs and the derivative markets. Additionally, with these ETF comes the options to short Bitcoin, making a potential profit as the value of Bitcoin falls. Taking advantages of contractions and expansions equally. If the pattern holds, this may be an example of how to apply the Elliot Wave Theory of investing, following the repeating patterns of ups and downs (Chen, 2023).

However, as more attention is being drawn to this cyclical pattern of Bitcoin, will it hold?

Bibliography

All-In Podcast. (2024, May 31). *Trump verdict, COVID Cover-up, Crypto Corner, Salesforce drops 20%, AI correction?* [Video]. YouTube. https://www.youtube.com/watch?v=R6hJh-OwoZw

Bitcoin price today, BTC to USD live price, marketcap and chart | CoinMarketCap. (n.d.). CoinMarketCap. https://coinmarketcap.com/currencies/bitcoin/historical-data/

Chen, J. (2023, September 8). *Elliott Wave Theory: What It Is and How to Use It.* Investopedia. https://www.investopedia.com/terms/e/elliottwavetheory.asp

Conway, L. (2024, August 8). *Bitcoin Halving: What It Is and Why It Matters for Crypto Investors.* Investopedia. https://www.investopedia.com/bitcoin-halving-4843769