Bitcoin Price Changes: A Regression Analysis

Adam Levin

13 December 2022

Introduction

For the final project, I hope to investigate the relationship between daily percent changes in Bitcoin (BTC) price and daily percent changes in the S&P 500. I would like to evaluate whether changes in the S&P 500 price could be utilized as a predictor of changes in Bitcoin price. My intuition is that there is likely to be a positive relationship between these price changes. I think this may be the case because, in recent years, cryptocurrencies have come to be traded in large quantities at the global scale (as the S&P 500 has been for many years). So, I would expect that the macroeconomic factors that influence price increases/decreases in the S&P 500 may influence similar increases/decreases in the price of Bitcoin, and thus that these respective daily price changes may align.

That said, I am cautious to predict that I will ultimately find a strong relationship between these price changes. After all, I have seen periods of time when Bitcoin has rapidly risen/fallen quite dramatically in comparison to the S&P 500, due to a number of factors. Nonetheless, I think this will be a fascinating relationship to explore and it will be interesting to see what other predictors of Bitcoin price change I am able to find in my analysis.

Processing of Data

To begin, I process the data that I will utilize in my analysis. I have downloaded two '.csv' files that contain the (1) BTC and (2) S&P 500 pricing data over the last decade or so.

I first read the two .csv files into data frames (with custome `colnames`) and merge them, then observe the head of and a summary of our merged data frame to evaluate the data we have at hand:

```
# read in dfs
bitcoin_df <- read.csv('Bitcoin.csv')
sp_df <- read.csv('S&P 500.csv')

# customize colnames
base_colnames <- c('price', 'open_price', 'hi_price', 'lo_price',
                   'vol_traded','price_pct_change')
colnames(bitcoin_df) <- c('date', paste('b_', base_colnames, sep=''))
colnames(sp_df) <- c('date', paste('sp_', base_colnames, sep=''))

df <- merge(bitcoin_df, sp_df) # merge
head(df) # observe head
```

```
##      date   b_price b_open_price b_hi_price b_lo_price b_vol_traded
## 1 1/10/11       0.3          0.3        0.3        0.3       10.36K
```

```
## 2 1/10/12       6.4          6.3        6.9          6      110.21K
## 3 1/10/13      14.1         13.8       14.3       13.8       51.81K
## 4 1/10/14     957.8          937      964.8      905.9        6.35K
## 5 1/10/17     904.4        899.8      911.3      890.1       53.71K
## 6 1/10/18 15,043.00    14,778.50  15,045.40  13,613.10       87.07K
##   b_price_pct_change sp_price sp_open_price sp_hi_price sp_lo_price
## 1              0.00% 1,269.75      1,270.84    1,271.52    1,262.18
## 2              0.00% 1,292.08      1,280.77    1,296.46    1,280.77
## 3              2.69% 1,472.12      1,464.64    1,472.30    1,461.02
## 4              2.22% 1,842.37      1,840.06    1,843.15    1,832.43
## 5              0.51% 2,268.90      2,269.72    2,279.27    2,265.27
## 6              1.79% 2,748.23      2,745.55    2,750.80    2,736.06
##   sp_vol_traded sp_price_pct_change
## 1            NA              -0.14%
## 2            NA               0.89%
## 3            NA               0.76%
## 4            NA               0.23%
## 5            NA               0.00%
## 6            NA              -0.11%
```

```
summary(df) # observe summary
```

```
##      date             b_price          b_open_price        b_hi_price
##  Length:3044        Length:3044        Length:3044        Length:3044
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##   b_lo_price        b_vol_traded       b_price_pct_change   sp_price
##  Length:3044        Length:3044        Length:3044        Length:3044
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##  sp_open_price      sp_hi_price        sp_lo_price        sp_vol_traded
##  Length:3044        Length:3044        Length:3044        Mode:logical
##  Class :character   Class :character   Class :character   NA's:3044
##  Mode  :character   Mode  :character   Mode  :character
##  sp_price_pct_change
##  Length:3044
##  Class :character
##  Mode  :character
```

I first observe that the `sp_vol_traded` column contains all null values (whereas none of the other columns contain any nulls), so we will elect to drop this column from our data frame. Further, I observe that each column of the data frame is of class `character`. For our purposes, I would like to convert the `date` column to the `Date` class so we can easily sort our data. Further, I would like to convert the other columns to the `numeric` class since they each represent potential numbers of interest. As the base `as.numeric` function cannot account for the ',', 'K', 'M', 'B', and '%' characters in some areas, I write a custom implementation to convert these to the class `numeric` (to have non-null data, we also drop the 5 out of 3044 rows for which the `b_vol_traded` column has no recording, i.e. a value of '-'). I make these changes below and create a new summary of the data:

```
df <- subset(df, select=-sp_vol_traded) # drop null `sp_vol_traded` column
df$date <- as.Date(df$date, format='%m/%d/%y') # convert `date` column to `Date` class
df <- df[df$b_vol_traded != '-',] # drop rows w/ null `b_vol_traded` values
df$b_vol_traded <- as.numeric(
```

```
  sub(
    '\\.', '', sub(
      'K', '0', sub(
        'M', '0000', sub(
          'B', '0000000', df$b_vol_traded
        )
      )
    )
  )
) # convert `b_vol_traded` column to numeric
df <- cbind(
  df[, c('date', 'b_vol_traded')],
  apply(
    df[, setdiff(colnames(df), c('date', 'b_vol_traded'))],
    2,
    function(c) as.numeric(sub('[,%]', '', c))
  )
) # convert rest of columns to numeric
df <- df[order(df$date),] # sort df by `date` in ascending order
rownames(df) <- NULL # reset index column of df
summary(df) # observe summary
```

```
##       date             b_vol_traded          b_price         b_open_price
##  Min.   :2010-10-04   Min.   :2.600e+02   Min.   :    0.1   Min.   :    0.1
##  1st Qu.:2013-10-16   1st Qu.:3.706e+04   1st Qu.:  157.8   1st Qu.:  160.2
##  Median :2016-10-21   Median :7.735e+04   Median :  802.0   Median :  795.0
##  Mean   :2016-10-22   Mean   :1.660e+07   Mean   : 8669.5   Mean   : 8666.9
##  3rd Qu.:2019-10-29   3rd Qu.:2.110e+05   3rd Qu.: 9301.0   3rd Qu.: 9306.5
##  Max.   :2022-11-03   Max.   :4.470e+09   Max.   :67527.9   Max.   :67528.7
##    b_hi_price        b_lo_price      b_price_pct_change    sp_price
##  Min.   :    0.1   Min.   :    0.0   Min.   :-57.2100   Min.   :1099
##  1st Qu.:  181.7   1st Qu.:  145.1   1st Qu.: -1.3650   1st Qu.:1729
##  Median :  833.1   Median :  771.0   Median :  0.0000   Median :2177
##  Mean   : 8921.3   Mean   : 8380.2   Mean   :  0.4982   Mean   :2464
##  3rd Qu.: 9481.2   3rd Qu.: 9131.0   3rd Qu.:  2.1400   3rd Qu.:2955
##  Max.   :68990.6   Max.   :66334.9   Max.   :336.8400   Max.   :4797
##  sp_open_price   sp_hi_price    sp_lo_price   sp_price_pct_change
##  Min.   :1097   Min.   :1125   Min.   :1075   Min.   :-11.98000
##  1st Qu.:1725   1st Qu.:1732   1st Qu.:1717   1st Qu.: -0.37000
##  Median :2177   Median :2183   Median :2170   Median :  0.06000
##  Mean   :2463   Mean   :2477   Mean   :2449   Mean   :  0.04513
##  3rd Qu.:2957   3rd Qu.:2976   3rd Qu.:2945   3rd Qu.:  0.55500
##  Max.   :4805   Max.   :4819   Max.   :4780   Max.   :  9.38000
```

Finally, let's add an indicator variable, `bear_bull`, that equals 1 during a Bull market (`date` <= '2020-02-19', and '2020-03-23' <= `date` <= '2022-01-03'), and 0 otherwise/during a Bear market (as per https://www.yardeni.com/pub/sp500corrbeartables.pdf). Note: here, we define a Bull market as a "a rise of 20% or more in a broad market index over at least a two-month period," and a Bear market as the reverse:

```
df$bear_bull <- 1*(
  df$date <= '2020-02-19' | (('2020-03-23' <= df$date) & (df$date <= '2022-01-03'))
) # construct `bear_bull` indicator column
```

Now that I have properly processed the data, let's discuss it!

Description of Data

The main data frame with which I will be working has $n = 3039$ rows that correspond to the days between October 4th, 2010 and November 3rd, 2022 for which we had non-null data. In total, there are 13 columns (including the `date` column). For both BTC and the S&P 500, I have access to the opening price of the day (`b_open_price` and `sp_open_price`), the low price of the day (`b_lo_price` and `sp_lo_price`), the high price of the day (`b_hi_price` and `sp_hi_price`), the final price of the day (`b_price` and `sp_price`), and the percent change in price over the course of the day (`b_price_pct_change` and `sp_price_pct_change`). Further, for Bitcoin, I have access to the volume traded in the day (`b_vol_traded`).

As mentioned above, the dependent variable I plan to use in my analysis is `b_price_pct_change` and the primary independent variable of interest will be `sp_price_pct_change`. Naturally, my investigation of this data will employ a longitudinal design since it considers data collected daily over a period of time.

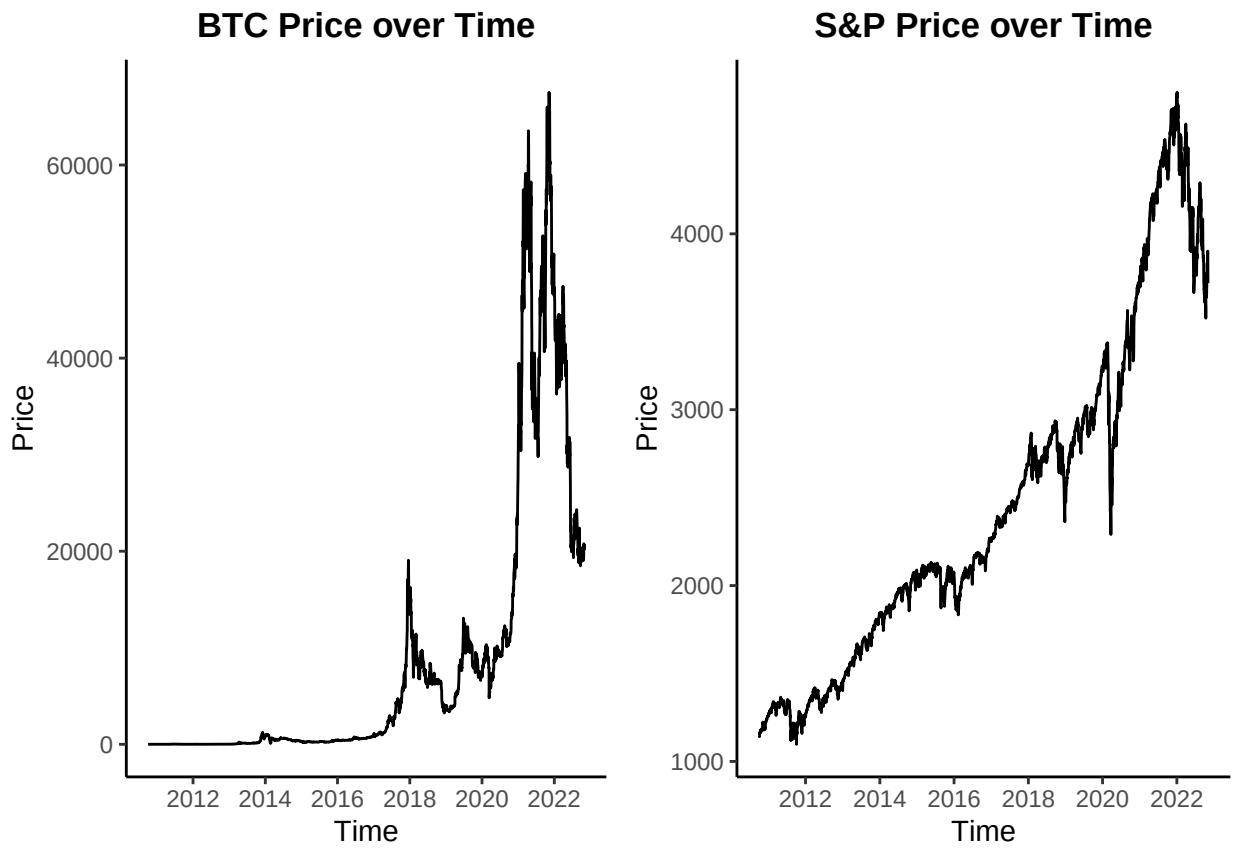
In addition to `sp_price_pct_change`, there are several other raw variables that could be interesting to include as independent variables in my analysis, as they too could have some underlying relationship with `b_price_pct_change`. It may be that Bitcoin price percent changes have some relationship with starting prices (of either Bitcoin or S&P 500), trading volume, and or whether the market is a Bear or a Bull. As such, I plan to include the `b_open_price`, `sp_open_price`, `b_vol_traded`, and `bear_bull` variables in a multiple regression analysis. Let's construct another data frame for our model, `indices`, containing these specific variables of interest:

```
indices <- df[
  ,
  c(
    'b_price_pct_change', 'sp_price_pct_change', 'date', 'b_open_price',
    'sp_open_price', 'b_vol_traded', 'bear_bull'
  )
] # construct indices
```

Preliminary Analysis of Data

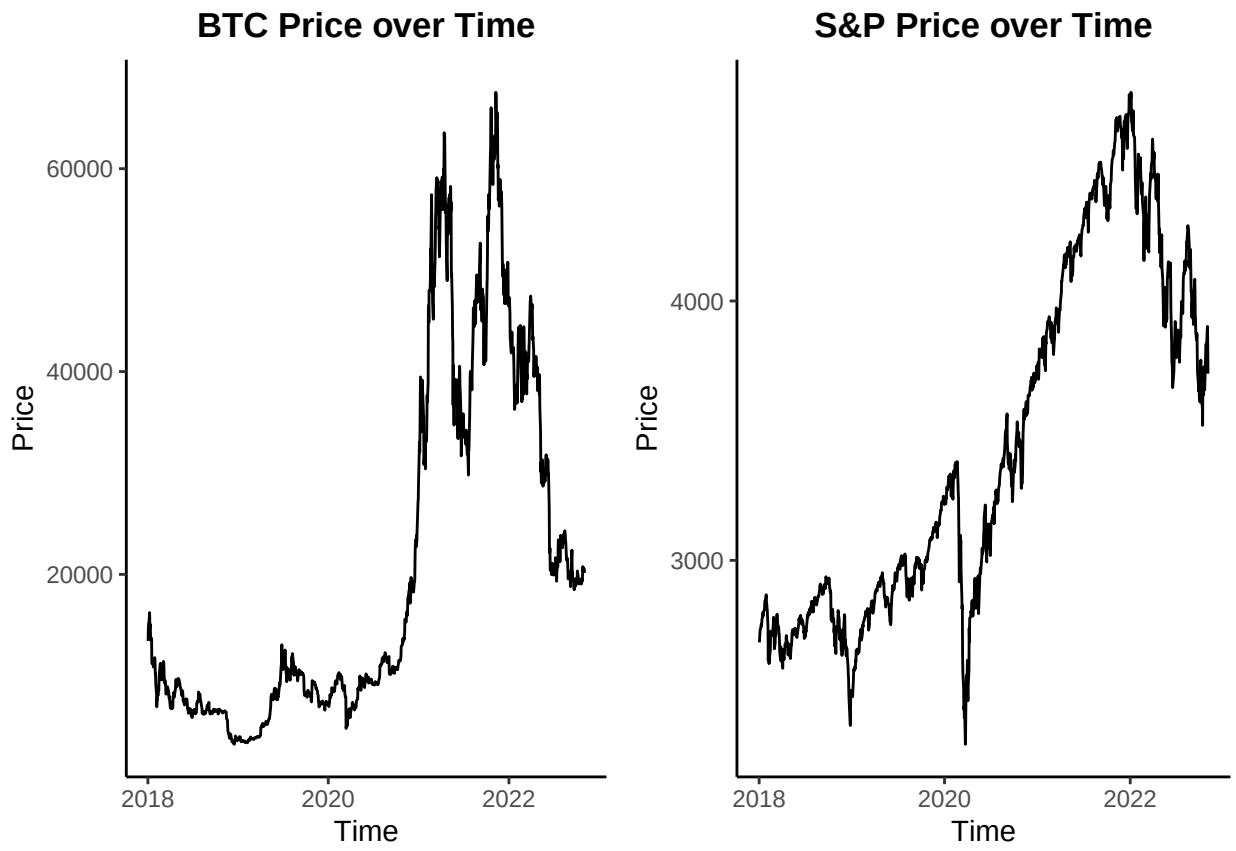
To begin, let's take a look at how the `b_open_price` and `sp_open_price` variables have changed over time:

```
# plot `b_price` and `sp_price` over time, side-by-side
library(gridExtra)
library(tidyverse)

plot1 <- ggplot(data = indices, mapping = aes(x = date, y = b_open_price)) +
  geom_line() + labs(title = "BTC Price over Time", x = "Time", y = "Price") +
  theme_classic() + theme(plot.title = element_text(face = "bold", hjust = 0.5))

plot2 <- ggplot(data = indices, mapping = aes(x = date, y = sp_open_price)) +
  geom_line() + labs(title = "S&P Price over Time", x = "Time", y = "Price") +
  theme_classic() + theme(plot.title = element_text(face = "bold", hjust = 0.5))

grid.arrange(plot1, plot2, ncol=2)
```



I quickly observe that BTC prices were extremely low compared to their present value until roughly the start of 2018. As such, I believe it makes sense to use only data available starting in 2018 for my analysis. As stated above, my intuition is that BTC prices may track well with the S&P 500 in recent years since there has been a notable worldwide market for BTC, so by subsetting just 2018 data and beyond, I will be able to zero in on a potential trend during this timeline. Let's construct a new data frame aligning with these rows and again examine them side-by-side:

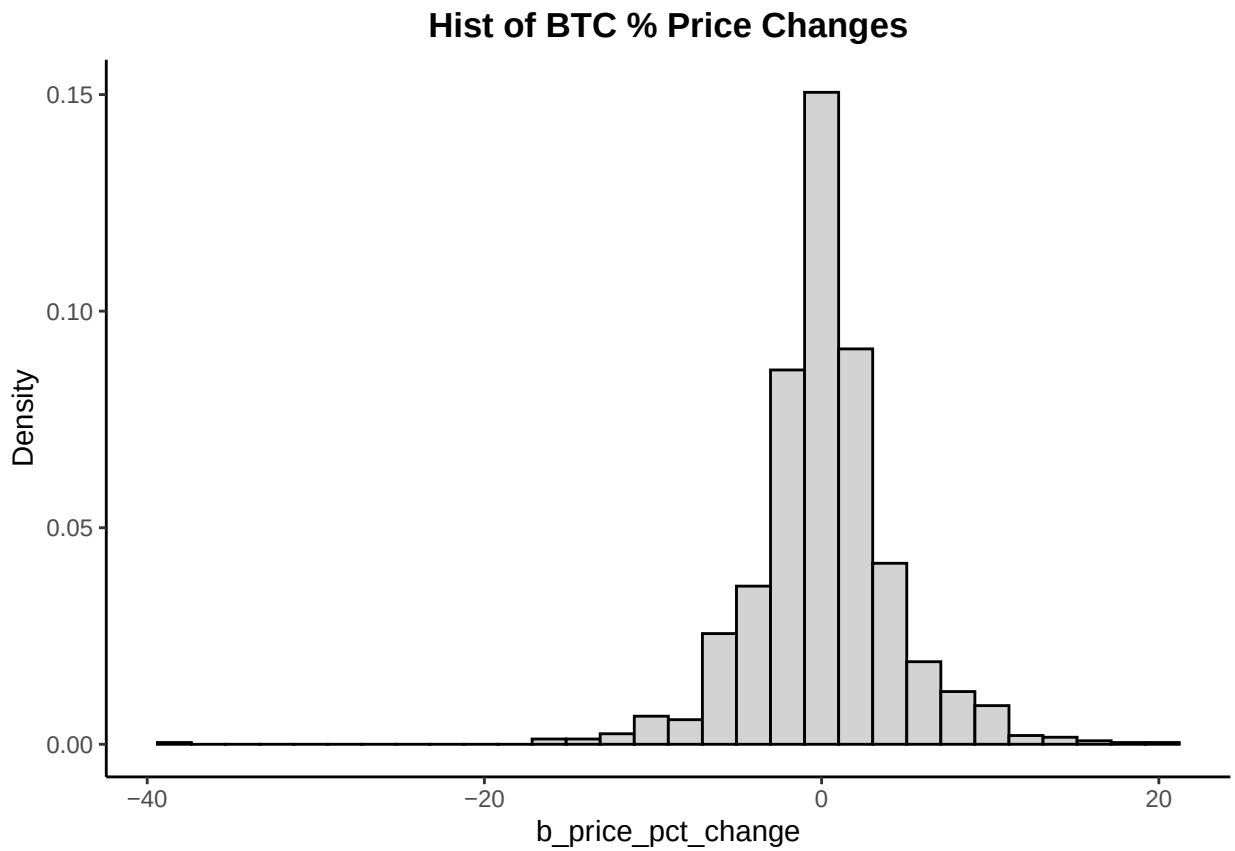
```
indices <- indices[indices$date >= '2018-01-01',] # subset `date` since 2018
rownames(indices) <- NULL # reset index column of indices

# plot `b_price` and `sp_price` over time since 2018, side-by-side
plot1 <- ggplot(data = indices, mapping = aes(x = date, y = b_open_price)) +
  geom_line() + labs(title = "BTC Price over Time", x = "Time", y = "Price") +
  theme_classic() + theme(plot.title = element_text(face = "bold", hjust = 0.5))
plot2 <- ggplot(data = indices, mapping = aes(x = date, y = sp_open_price)) +
  geom_line() + labs(title = "S&P Price over Time", x = "Time", y = "Price") +
  theme_classic() + theme(plot.title = element_text(face = "bold", hjust = 0.5))

grid.arrange(plot1, plot2, ncol=2)
```



Our new data set has $n = 1220$ rows. We observe that BTC price remains fairly level until late 2020, after which it takes a steep climb into early 2021, followed by a significant drop and climb toward the start of 2022, after which it has seen a notable decline. Meanwhile, the S&P steadily climbed until a sharp drop in early 2020 (around the time COVID hit), followed by a steady climb until late 2021 and a fairly steady decline throughout 2022. In general, it appears that these trends most closely track since mid-2021, so it could be interesting to later compare our since-2018 analysis to our since mid-2021 analysis to see if the respective price changes have followed a particularly strong relationship in the last 18 months or so.

Having subsetted our data to 2018 and beyond, let's construct a histogram of `b_price_pct_change` to examine the distribution of our outcome/dependent variable:

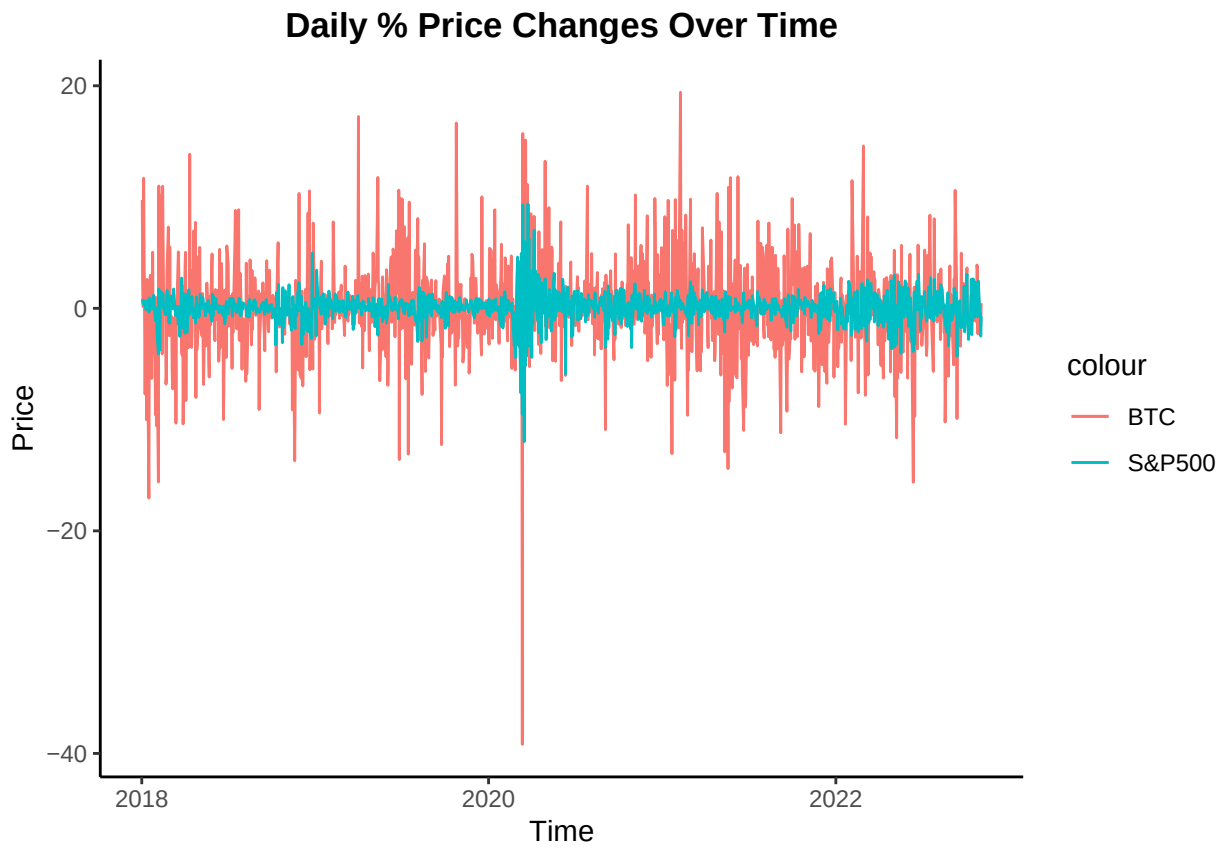
```
Histogram <- indices[indices$date >= '2018-01-01',] |>
ggplot(mapping = aes(x = b_price_pct_change)) +
geom_histogram(mapping = aes(y = ..density..), color = "black", fill = "light gray") +
  theme_classic() + labs(title = "Hist of BTC % Price Changes",
                         x = "b_price_pct_change", y = "Density") + theme_classic() +
  theme(plot.title = element_text(face = "bold", hjust = 0.5))

Histogram
```



We see that the daily % price changes in BTC take a unimodal, roughly Bell-shaped curve and thus follow an approximately Normal distribution centered around 0%, with the great majority of values falling between -10% and 10%. Notably, there exist outliers, particularly one extreme outlier which falls near -40%, whereas no other daily change falls below -20%.

For further context, let's also take a look at how `b_price_pct_change` (our dependent variable) and `sp_price_pct_change` (our primary independent variable) have tracked over this time period:

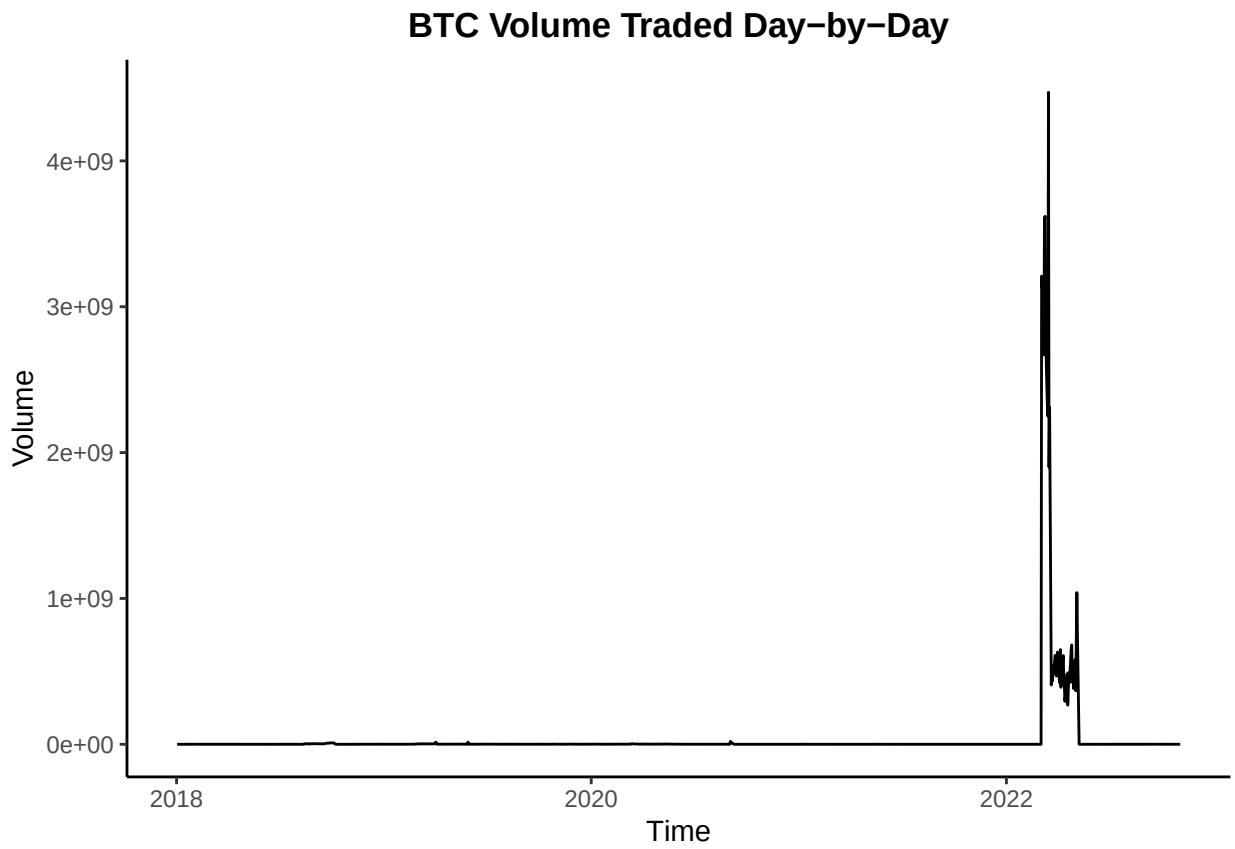
```
# plot BTC % daily price changes over time and overlay S&P 500 % daily price changes
ggplot(data = indices, mapping = aes(x = date)) + geom_line(aes(y = b_price_pct_change,
                                                                col = "BTC")) +
  geom_line(aes(y = sp_price_pct_change, col = "S&P500")) +
  labs(title = "Daily % Price Changes Over Time", x = "Time", y = "Price") +
  theme_classic() + theme(plot.title = element_text(face = "bold", hjust = 0.5))
```



Day-by-day, the percent price changes in both BTC and the S&P 500 are rather stochastic/jumpy. Generally, it appears that BTC experiences more dramatic % changes in price day by day, as evidenced by the fact that the red line tracking its changes typically jumps to farther distances from the 0% change line in either direction. That said, the S&P did appear to experience notable day-by-day price changes in early 2020 around the time COVID hit. By inspection, it appears that jumps and falls in BTC prices day-by-day at least somewhat align with jumps and falls in S&P prices day-by-day. That said, it is not possible to deduce the strength of this potential relationship from this plot alone.

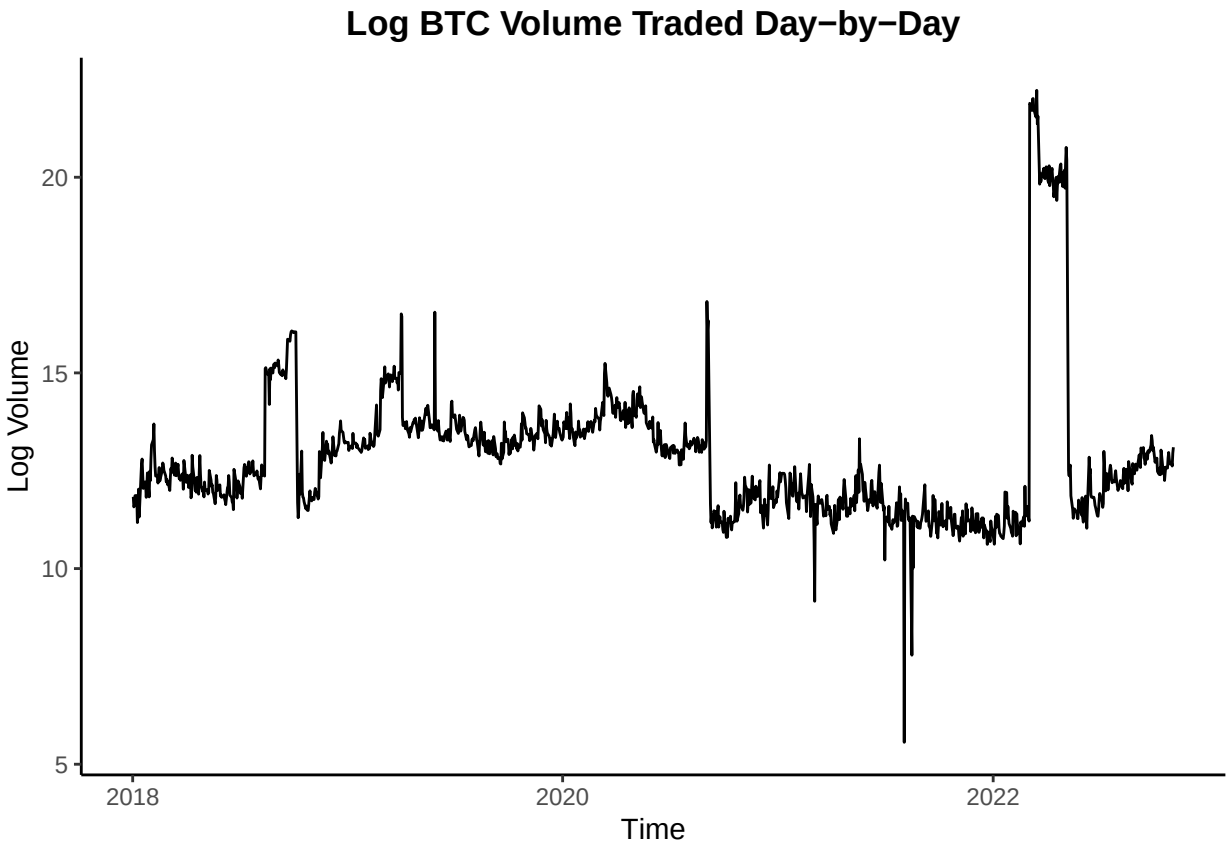
Finally, let's take a look at how the `b_vol_traded` changes over time to get a further glance into this variable of interest:

```
# plot `b_vol_traded` over time
ggplot(data = indices, mapping = aes(x = date, y = b_vol_traded)) + geom_line() +
  labs(title = "BTC Volume Traded Day-by-Day", x = "Time", y = "Volume") +
  theme_classic() + theme(plot.title = element_text(face = "bold", hjust = 0.5))
```



Seeing how there are notably extreme volumes traded in early 2022 that far trump the volumes traded during any other span of time, let's replace this column with a log transformation and re-evaluate:

```
indices$log_b_vol_traded <- log(indices$b_vol_traded) # add log transformed column
indices <- subset(indices, select=-b_vol_traded) # drop `b_vol_traded` column

# plot `log_b_vol_traded` over time
ggplot(data = indices, mapping = aes(x = date, y = log_b_vol_traded)) + geom_line() +
  labs(title = "Log BTC Volume Traded Day-by-Day", x = "Time", y = "Log Volume") +
  theme_classic() + theme(plot.title = element_text(face = "bold", hjust = 0.5))
```

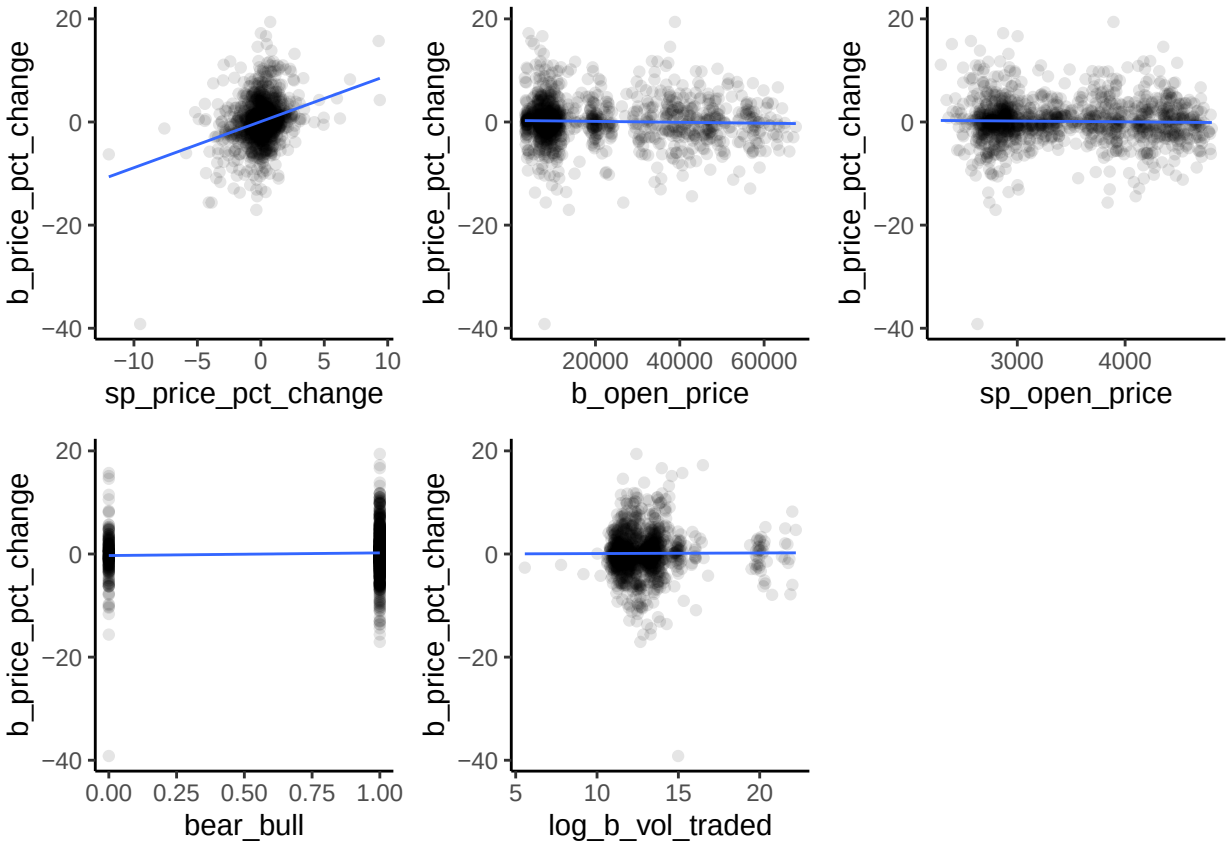


On the log-scale, we see a general trend of increase in volume through mid-2020 (including a few sharp jumps), a notable drop in late 2020, followed by relative consistency in log-volume thereafter (barring a notable drop/recovery in mid-2021 and a sharp increase/ensuing drop in early 2022).

Now that we have considered each of the variables with respect to time, let's examine scatterplots of our outcome variable plotted against each of our independent variables to take an initial glance at their respective relationships (along with fit lines based on their respective single variable linear regressions):

```
# plot `b_price_pct_change` versus each independent variable
plot1 <- ggplot(data = indices, mapping = aes(x = sp_price_pct_change,
                                              y = b_price_pct_change)) +
  geom_point(alpha = .1) + labs(x = "sp_price_pct_change",
                                y = "b_price_pct_change") +
  geom_smooth(method = lm, alpha = 0, size = .5) + theme_classic()

plot2 <- ggplot(data = indices, mapping = aes(x = b_open_price,
                                              y = b_price_pct_change)) +
  geom_point(alpha = .1) + labs(x = "b_open_price",
                                y = "b_price_pct_change") +
  geom_smooth(method = lm, alpha = 0, size = .5) + theme_classic()

plot3 <- ggplot(data = indices, mapping = aes(x = sp_open_price,
                                              y = b_price_pct_change)) +
  geom_point(alpha = .1) + labs(x = "sp_open_price",
                                y = "b_price_pct_change") +
  geom_smooth(method = lm, alpha = 0, size = .5) + theme_classic()
```

```
plot4 <- ggplot(data = indices, mapping = aes(x = bear_bull,
                                              y = b_price_pct_change)) +
  geom_point(alpha = .1) + labs(x = "bear_bull",
                                y = "b_price_pct_change") +
  geom_smooth(method = lm, alpha = 0, size = .5) + theme_classic()

plot5 <- ggplot(data = indices, mapping = aes(x = log_b_vol_traded,
                                              y = b_price_pct_change)) +
  geom_point(alpha = .1) + labs(x = "log_b_vol_traded",
                                y = "b_price_pct_change") +
  geom_smooth(method = lm, alpha = 0, size = .5) + theme_classic()


grid.arrange(plot1, plot2, plot3, plot4, plot5, ncol = 3, nrow = 2)
```



By inspection, there appears to be a positive relationship between `b_price_pct_change` and `sp_price_pct_change`—our main independent variable of interest—as higher values of `b_price_pct_change` tend to coincide with higher values of `sp_price_pct_change`. It does not appear that `b_price_pct_change` shares much of a notable relationship with any of the other four independent variables of interest. It will be interesting to see whether this relationship between `b_price_pct_change` and `sp_price_pct_change` proves to be statistically significant in our next section.

	Model 1
(Intercept)	−2.570
	s.e. = 1.861
	p = 0.168
sp_price_pct_change	0.888
	s.e. = 0.085
	p = <0.001
b_open_price	−0.000 03
	s.e. = 0.000 02
	p = 0.068
sp_open_price	0.0007
	s.e. = 0.0005
	p = 0.124
bear_bull	0.465
	s.e. = 0.377
	p = 0.218
log_b_vol_traded	0.036
	s.e. = 0.067
	p = 0.594
Num.Obs.	1220
R2	0.087
R2 Adj.	0.083

Results

To investigate our research question, we will run a multiple linear regression, using the `sp_price_pct_change`, `b_open_price`, `sp_open_price`, `bear_bull`, and `log_b_vol_traded` variables as independent variables/predictors. Below, we run this regression and present a table of the results (seen above):

```
mr_model <- lm(b_price_pct_change ~ sp_price_pct_change + b_open_price +
                 sp_open_price + bear_bull + log_b_vol_traded, indices)

sp_price_pct_change_CI <- confint(mr_model, level = .95)['sp_price_pct_change',]

modelsummary::modelsummary(mr_model,
statistic = c("s.e. = {std.error}",
"p = {p.value}"),
gof_map = c("nobs", "r.squared", "adj.r.squared"))
```

Keeping in mind that our outcome variable, `b_price_pct_change`, ranges from approximately -40% to 20%, whereas our primary independent variable of interest, `sp_price_pct_change`, ranges from approximately -20% to 10%, the model suggests that on average, a one unit increase in the daily S&P 500 % price change coincides with a 0.888 unit increase in the daily BTC % price change, when holding the other variables (`b_open_price`, `sp_open_price`, `bear_bull`, and `log_b_vol_traded`) constant. This coefficient for `sp_price_pct_change` (roughly 0.888) is statistically significant at the 0.01 level because the p-value is less than 0.001. However, I do not believe that this coefficient represents a causal effect. While it does suggest that there is a correlation between these two variables, as discussed above, the S&P and BTC are each traded throughout the day and their daily price changes likely result from a number of macroeconomic factors for which our model does not account. So, while it is interesting to have established a statistically significant relationship between these variables, we do not have sufficient reason to deduce that changes in S&P price cause changes in BTC price.

Based on the standard error of our `sp_price_pct_change` coefficient, we construct a 95% confidence interval for the `sp_price_pct_change` coefficient of [0.7201743, 1.0554715]. This means that, assuming that the sampling distribution of BTC % price changes is normally distributed (which seems to be a reasonable assumption based on the interpretation of our histogram above), we are 95% confident that a one unit increase in the S&P 500 % price change coincides with a 0.7201743 to a 1.0554715 unit increase in the % price change of Bitcoin (when holding the other independent variables in our model constant), which cannot be attributed to "random chance."

Notably, none of the coefficients for any of the other variables are statistically significant at the 0.01 or 0.05 level since each of their respective p-values came out greater than 0.05. Further, with an R-squared value of 0.087, we have that on average, the multiple regression model accounts for roughly 8.7% of the variation in the outcome `b_price_pct_change` variable. It's important to note that this is a rather low R-squared value, which suggests that the model does not have a strong goodness-of-fit and would likely struggle to produce precise predictions of the outcome `b_price_pct_change`.

The intercept of -2.57 suggests that on average, a day with values of 0 for each of the predictors (note that a 0 of `log_b_vol_traded` is a 1 of `b_vol_traded` due to the transformation) would have a BTC daily change in price of -2.57%. This is not a meaningful quantity to interpret since several of these predictors never have values close to 0 in our data (for instance, the minimum value of `b_open_price` in our data is 3248.3).

[Note that we have already presented a scatterplot of the outcome and the main independent variable in the previous section above.]

Conclusion

In this study, we aimed to examine the relationship between daily changes in the S&P 500 price and corresponding daily changes in the price of BTC. In our primary analysis, we performed a multiple linear regression analysis with our outcome variable being the daily percent changes in BTC price, and our independent variables including the daily percent changes in the S&P 500 price, the daily opening prices of both the S&P and BTC, an indicator of whether the market was defined as Bear or Bull for the day (as given in the source mentioned above), and the log of the daily volume traded of BTC. Ultimately, we only found a statistically significant coefficient for the % change in S&P price variable: we found that, when holding the other variables constant, a one unit increase in the daily S&P % price change coincided with 0.888 unit increase in the daily BTC % price change, on average (with a 95% confidence interval ranging from a 0.7201743 to a 1.0554715 unit increase in daily BTC % price change).

One limitation of our study is that our multiple regression analysis produced a rather low R-squared value of 0.087, likely suggesting that our model would struggle to produce precise predictions of the outcome `b_price_pct_change`. Another limitation is that we restricted our model to a linear regression, when there could potentially exist nonlinear relationships between some of our independent variables and the outcome. Another major limitation of our study is that we only considered five rather basic variables as predictors in our model (each of which we derived from raw S&P 500 and BTC daily pricing data). Various other factors beyond the data we considered could influence daily changes in BTC price, such as changes in interest rates, changes in exchange rates, changes in crypto or monetary exchanges that hold concentrated amounts of BTC (and downfalls of those exchanges), and integration of BTC into payment platforms and institutional financial market makers (such as Visa/Citadel), for instance.

To improve the study in the future, it would be interesting to attempt to account for some of these limitations. For instance, it could be enticing to move beyond linear regression, and perhaps utilize higher-level statistical methods that could account for potential nonlinear relationships in the data. It could also be useful to see what data we could attain regarding several of the macroeconomic factors above that may affect changes in Bitcoin price, and incorporate that data into our analysis. Finally, it could be interesting to see if previous changes in the S&P 500 could be used to predict changes in BTC price. Our current model uses the same-day % changes in the S&P 500 as a predictor of the changes in BTC price, but it might be interesting to incorporate some sort of moving averages as predictors (such as the average % changes in both BTC and the

S&P 500 over the previous 10 days, for instance), to see if recent previous changes in these variables might have statistically significant effects on the daily % changes in BTC price.