SEC1-SA2-GROUP9-AGUILA,V-MASICAT,L-SA2

2024-05-12

Bitcoin Trading

Find out which probability distribution function best fits Bitcoin's returns for trading data every minute, from January 1, 2012 to April 15, 2024, for Bitcoin quoted in United States dollars or the BTC/USD pair.

```r
# Define Tsallis distribution function
rtsallis <- function(n, q, mean, sd) {
  x <- rnorm(n, mean, sd)
  x <- x / sqrt(q)
  return(x)
}

# Load necessary libraries
library(stats)

# Load the CSV file
df <- read.csv("/Users/veraaguila/Downloads/btcusd.csv")
head(df, 10)
```

```
##         Date     Price      Open      High       Low   Vol. Change..
## 1   9-Feb-24 47,545.40 45,293.30 47,710.20 45,254.20 86.85K    4.97%
## 2   8-Feb-24 45,293.30 44,346.20 45,579.20 44,336.40 66.38K    2.15%
## 3   7-Feb-24 44,339.80 43,088.40 44,367.90 42,783.50 48.57K    2.91%
## 4   6-Feb-24 43,087.70 42,697.60 43,375.50 42,566.80 33.32K    0.91%
## 5   5-Feb-24 42,697.20 42,581.40 43,532.20 42,272.50 39.26K    0.27%
## 6   4-Feb-24 42,581.40 43,006.20 43,113.20 42,379.40 20.33K   -0.99%
## 7   3-Feb-24 43,005.70 43,194.70 43,370.40 42,882.00 14.57K   -0.44%
## 8   2-Feb-24 43,194.70 43,083.70 43,459.30 42,596.30 42.65K    0.26%
## 9   1-Feb-24 43,081.40 42,580.10 43,263.10 41,890.50 47.69K    1.18%
## 10 31-Jan-24 42,580.50 42,946.20 43,739.70 42,315.40 56.48K   -0.85%
```

```r
# Convert 'Price' column to numeric
df$Price <- as.numeric(as.character(df$Price))
```

```
## Warning: NAs introduced by coercion
```

```r
# Create a function to drop NA values and calculate returns
DropNA <- function(x) {
  x <- diff(log(x$Price))
  x <- x[!is.na(x)]
  return(x)
}

# Create random samples for each distribution and perform Kolmogorov-Smirnov tests
distributions <- c("normal", "t", "cauchy", "tsallis", "powerlaw")
results <- data.frame(Distribution = distributions, D_statistic = numeric(length(distributions)))

for (dist in distributions) {
  set.seed(123) # Set seed for reproducibility
  df_teste <- switch(dist,
                     normal = rnorm(length(DropNA(df)), mean = mean(DropNA(df)), sd = sd(DropNA(df))),
                     t = rt(length(DropNA(df)), df = length(DropNA(df)) - 1),
                     cauchy = rcauchy(length(DropNA(df)), location = mean(DropNA(df)), scale = sd(DropNA(df))),
                     tsallis = rtsallis(length(DropNA(df)), q = 2, mean = mean(DropNA(df)), sd = sd(DropNA(df))),
                     powerlaw = poweRlaw::rplcon(length(DropNA(df)), -0.3, sd(DropNA(df))))


  ks_result <- ks.test(DropNA(df), df_teste)
  results[results$Distribution == dist, "D_statistic"] <- ks_result$statistic
}
```

```
## Warning in ks.test.default(DropNA(df), df_teste): p-value will be approximate
## in the presence of ties
```

```
## Warning in ks.test.default(DropNA(df), df_teste): p-value will be approximate
## in the presence of ties
## Warning in ks.test.default(DropNA(df), df_teste): p-value will be approximate
## in the presence of ties
## Warning in ks.test.default(DropNA(df), df_teste): p-value will be approximate
## in the presence of ties
## Warning in ks.test.default(DropNA(df), df_teste): p-value will be approximate
## in the presence of ties
```

```r
# Show the results
results
```

```
##   Distribution D_statistic
## 1       normal   0.2196846
## 2            t   0.4437194
## 3       cauchy   0.2452420
## 4      tsallis   0.1636759
## 5     powerlaw   0.7493203
```

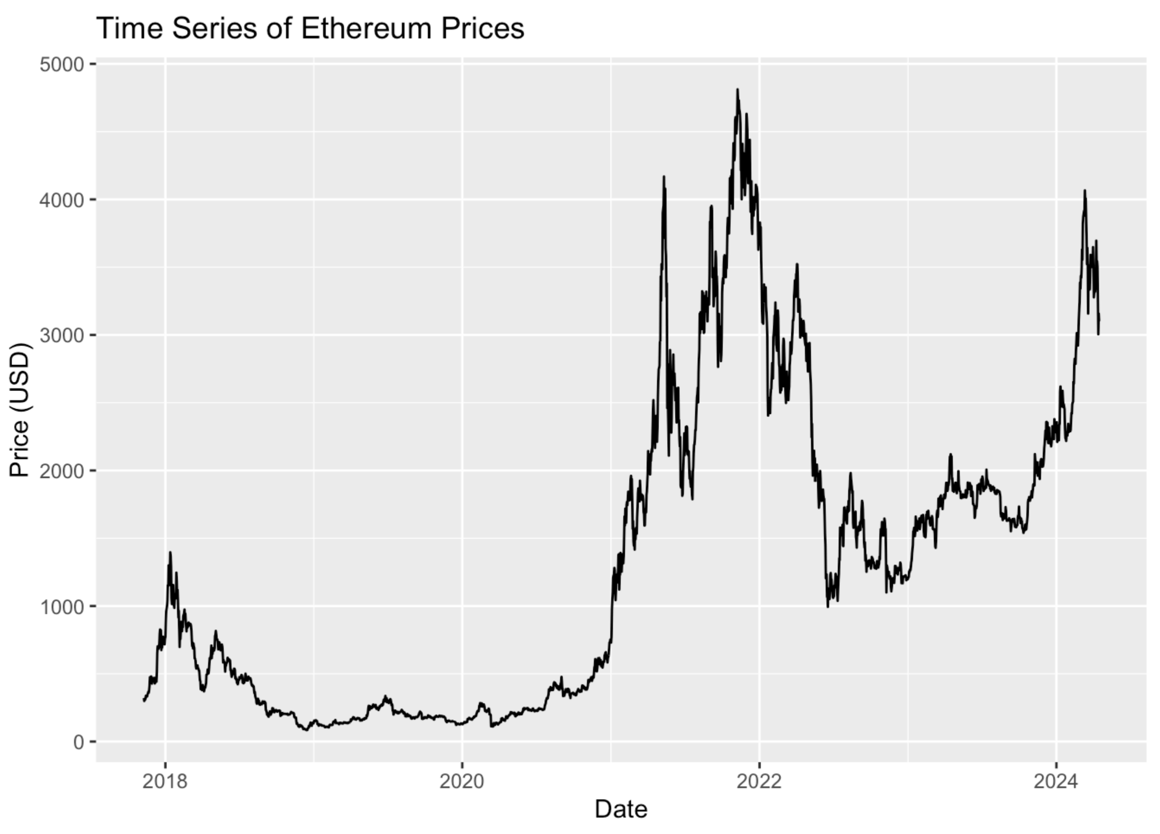
```r
# Find the distribution with the smallest D_statistic
best_distribution <- results[which.min(results$D_statistic), "Distribution"]
best_statistic <- min(results$D_statistic)

# Print the conclusion
cat("The best-fitting distribution for Bitcoin's returns is", best_distribution, "with a D_statistic of", best_st
atistic, "\n")
```

```
## The best-fitting distribution for Bitcoin's returns is tsallis with a D_statistic of 0.1636759
```

````


## Ethereum Trading
Test using Shapiro-Wilk normality test the Ethereum returns for trading data every five minutes, from August 7, 2
015 to April 15, 2024.







```r
ETHD <- na.omit(ETHD)

summary(ETHD)
````

```
##       Date                 Open              High              Low         
##  Min.   :2017-11-09   Min.   :  84.28   Min.   :  85.34   Min.   :  82.83  
##  1st Qu.:2019-06-19   1st Qu.: 239.02   1st Qu.: 244.21   1st Qu.: 232.45  
##  Median :2021-01-26   Median :1167.97   Median :1204.62   Median :1093.91  
##  Mean   :2021-01-26   Mean   :1315.76   Mean   :1353.18   Mean   :1274.64  
##  3rd Qu.:2022-09-05   3rd Qu.:1935.06   3rd Qu.:1987.60   3rd Qu.:1885.02  
##  Max.   :2024-04-15   Max.   :4810.07   Max.   :4891.70   Max.   :4718.04  
##      Close           Adj Close           Volume         
##  Min.   :  84.31   Min.   :  84.31   Min.   :6.217e+08  
##  1st Qu.: 238.95   1st Qu.: 238.95   1st Qu.:5.013e+09  
##  Median :1168.33   Median :1168.33   Median :9.699e+09  
##  Mean   :1316.75   Mean   :1316.75   Mean   :1.224e+10  
##  3rd Qu.:1936.58   3rd Qu.:1936.58   3rd Qu.:1.679e+10  
##  Max.   :4812.09   Max.   :4812.09   Max.   :8.448e+10
```

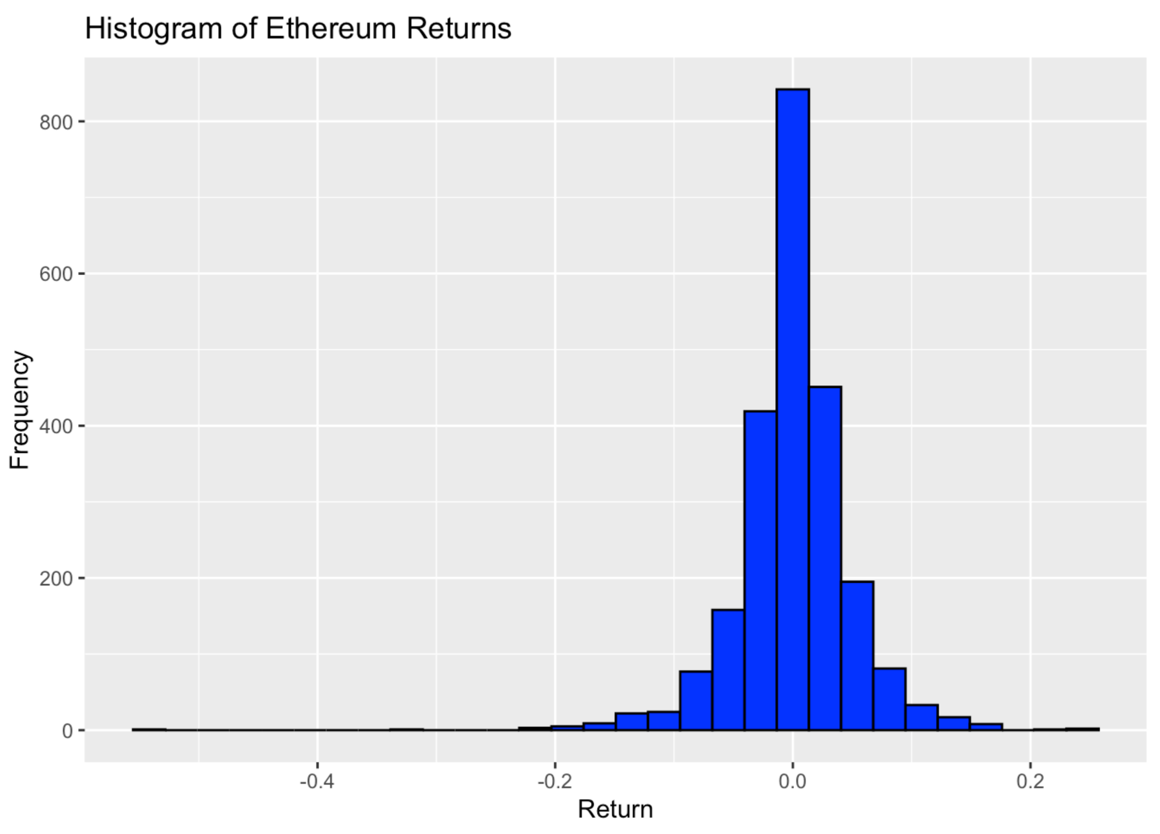
```r
ggplot(ETHD, aes(x = Date, y = Close)) +
  geom_line() +
  labs(x = "Date", y = "Price (USD)", title = "Time Series of Ethereum Prices")
```




```r
ETHD$Return <- c(NA, diff(log(ETHD$Close)))

summary(ETHD$Return)
```

```
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.       NA's 
## -0.5507317 -0.0188802  0.0008059  0.0009658  0.0237581  0.2347406          1
```

```r
ggplot(ETHD, aes(x = Return)) +
  geom_histogram(bins = 30, fill = "blue", color = "black", na.rm = TRUE) +
  labs(x = "Return", y = "Frequency", title = "Histogram of Ethereum Returns")
```




```r
ST <- shapiro.test(ETHD$Return)
print("Shapiro-Wilk Test:")
```

```
## [1] "Shapiro-Wilk Test:"
```

```r
print(ST)
```

```
## 
##  Shapiro-Wilk normality test
## 
## data:  ETHD$Return
## W = 0.91948, p-value < 2.2e-16
```

```r
alpha <- 0.05
if (ST$p.value > alpha) {
  print("The returns are normally distributed (fail to reject H0)")
} else {
  print("The returns are not normally distributed (reject H0)")
}
```

```
## [1] "The returns are not normally distributed (reject H0)"
```

```r
summTable <- ETHD %>%
  summarize(
    MeanReturn = mean(Return, na.rm = TRUE),
    SDReturn = sd(Return, na.rm = TRUE),
    MinReturn = min(Return, na.rm = TRUE),
    MaxReturn = max(Return, na.rm = TRUE)
  )
kable(summTable)
```

MeanReturn	SDReturn	MinReturn	MaxReturn
0.0009658	0.0474936	-0.5507317	0.2347406
```