SA2

Dizon, Kiana Marie & Delos Santos, Paul Joaquin

2024-05-12

Summative Assessment 2

1. 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.

```
# 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("C:/Users/Kiana/Documents/R Language/SA2/bitstampUSD_1-min_data_2012-0
1-01_to_2024-02-09.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%
```

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

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

```
# 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(dist
ributions)))

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(d
f)), 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
```

```
# 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
```

```
# 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, "wit
h a D_statistic of", best_statistic, "\n")
```

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