# SEC-1-SA2-GROUP-2-SIGUE,-JP, QUIJANO, JP-SA2

Sigue, John Patrick A., Quijano, Julian Philip S.

2025-05-16

## 1. Find out which probability distribution function best fits Bitcoin's returns for trading data every minute, from January 1, 2012 to April 15, 2025, for Bitcoin quoted in United States dollars or the BTC/USD pair.

### Load the data

```
# Load CSV
data <- read.csv("btcusd_1-min_data.csv")

# Convert timestamp
data$Timestamp <- as.POSIXct(data$Timestamp, origin = "1970-01-01", tz = "UTC")

# Limit only until April 15, 2025
data <- data[data$Timestamp <= as.POSIXct("2025-04-15 23:59:59", tz = "UTC"),
    ]

# Keep only Timestamp and Close columns
clean <- data[, c("Timestamp", "Close")]

# Calculate Simple Returns
clean$SimpleReturn <- c(NA, (diff(clean$Close)/head(clean$Close, -1)))

# Remove NA
clean <- na.omit(clean)
```

### Summary of Data

```
# Summary
summary(clean)
```

```
##    Timestamp                       Close           SimpleReturn
##  Min.   :2012-01-01 10:02:00.00   Min.   :     3.8   Min.   :-0.4604705
##  1st Qu.:2015-04-28 14:41:15.00   1st Qu.:   423.2   1st Qu.:-0.0002378
##  Median :2018-08-23 19:20:30.00   Median :  6556.0   Median : 0.0000000
##  Mean   :2018-08-23 19:27:57.39   Mean   : 17164.2   Mean   : 0.0000031
##  3rd Qu.:2021-12-18 23:59:45.00   3rd Qu.: 27093.0   3rd Qu.: 0.0002485
##  Max.   :2025-04-15 23:59:00.00   Max.   :109036.0   Max.   : 0.8532000
```

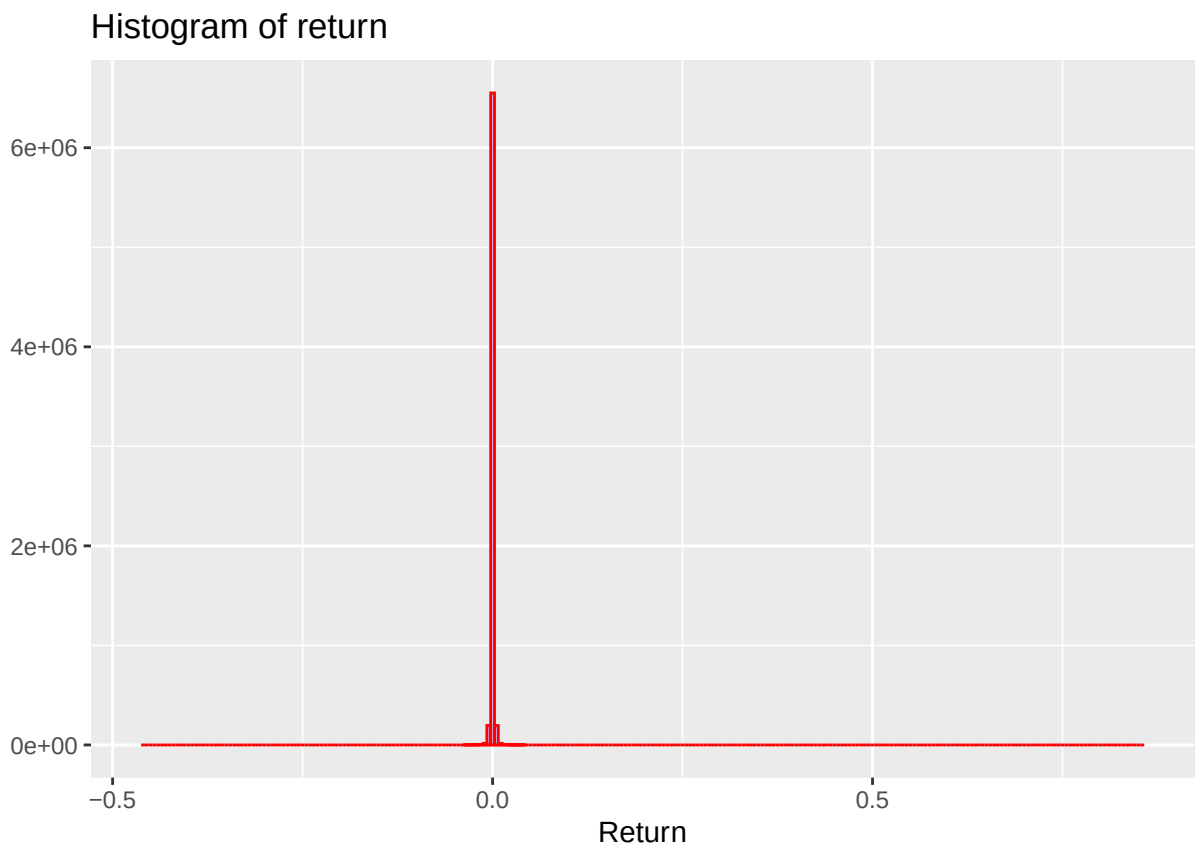
### Histogram of Data

```
# Create histogram of SimpleReturn
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.4.3
```

```
qplot(clean$SimpleReturn, geom = "histogram", binwidth = 0.005, main = "Histogram of return",
    xlab = "Return", fill = I("blue"), col = I("red"), alpha = I(0.2))
```

```
## Warning: `qplot()` was deprecated in ggplot2 3.4.0.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```




### Kolmogorov-Smirnov tests

The Kolmogorov-Smirnov (KS) test is a non-parametric statistical test used to compare a sample with a reference probability distribution or to compare two samples. It measures the largest absolute difference (D statistic) between the cumulative distribution function (CDF) of the sample and that of the theoretical or simulated distribution. A smaller D value indicates that the sample distribution is closer to the reference distribution. The KS test also provides a p-value, which assesses the statistical significance of the observed

difference — a high p-value (typically $> 0.05$) suggests that there is no strong evidence to reject the null hypothesis that both distributions are the same. In practice, especially with large datasets, even small differences can lead to very low p-values, so when identifying the distribution that best fits your data, the D statistic is more reliable for comparing closeness, while the p-value helps assess statistical significance.

```
sim_n <- rnorm(length(clean$SimpleReturn), mean = mean(clean$SimpleReturn),
    sd = sd(clean$SimpleReturn))

nor <- ks.test(clean$SimpleReturn, sim_n)
```

### Normal Distribution

```
## Warning in ks.test.default(clean$SimpleReturn, sim_n): p-value will be
## approximate in the presence of ties
```

```
print(nor)
```

```
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  clean$SimpleReturn and sim_n
## D = 0.21406, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

```
sim_t <- rt(length(clean$SimpleReturn), df = length(clean$SimpleReturn) -
    1)

stu <- ks.test(clean$SimpleReturn, sim_t)
```

### Student's T Distribution

```
## Warning in ks.test.default(clean$SimpleReturn, sim_t): p-value will be
## approximate in the presence of ties
```

```
print(stu)
```

```
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  clean$SimpleReturn and sim_t
## D = 0.49456, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

```r
library(rmutil)
```

### Laplace Distribution

```
##
## Attaching package: 'rmutil'
```

```
## The following object is masked from 'package:stats':
##
##     nobs
```

```
## The following objects are masked from 'package:base':
##
##     as.data.frame, units
```

```r
sim_laplace <- rlaplace(length(clean$SimpleReturn), m = mean(clean$SimpleReturn),
    s = sd(clean$SimpleReturn))

lap <- ks.test(clean$SimpleReturn, sim_laplace)
```

```
## Warning in ks.test.default(clean$SimpleReturn, sim_laplace): p-value will be
## approximate in the presence of ties
```

```r
print(lap)
```

```
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  clean$SimpleReturn and sim_laplace
## D = 0.20248, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

```r
library(poweRlaw)
```

### Power Law Distribution

```
## Warning: package 'poweRlaw' was built under R version 4.4.3
```

```r
set.seed(1)
sim_powerlaw <- rplcon(n = length(clean$SimpleReturn), xmin = 0.01, alpha = 2.5)

clean_positive <- clean$SimpleReturn[clean$SimpleReturn > 0]
sim_powerlaw <- sim_powerlaw[1:length(clean_positive)]

pow <- ks.test(clean_positive, sim_powerlaw)
```

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

```
print(pow)
```

```
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  clean_positive and sim_powerlaw
## D = 0.99559, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

```
sim_uniform <- runif(length(clean$SimpleReturn), min = min(clean$SimpleReturn),
    max = max(clean$SimpleReturn))

uni <- ks.test(clean$SimpleReturn, sim_uniform)
```

### Uniform Distribution

```
## Warning in ks.test.default(clean$SimpleReturn, sim_uniform): p-value will be
## approximate in the presence of ties
```

```
print(uni)
```

```
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  clean$SimpleReturn and sim_uniform
## D = 0.64066, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

```
returns_pos <- clean$SimpleReturn - min(clean$SimpleReturn) + 1e-06  # Make all values positive

sim_exp <- rexp(length(returns_pos), rate = 1/mean(returns_pos))

exp <- ks.test(clean$SimpleReturn, sim_exp)
```

### Exponential Distribution

```
## Warning in ks.test.default(clean$SimpleReturn, sim_exp): p-value will be
## approximate in the presence of ties
```

```
print(exp)
```

```
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  clean$SimpleReturn and sim_exp
## D = 0.98157, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

```
sim_cauchy <- rcauchy(length(clean$SimpleReturn), location = median(clean$SimpleReturn),
    scale = IQR(clean$SimpleReturn)/2)

cau <- ks.test(clean$SimpleReturn, sim_cauchy)
```

### Cauchy Distribution

```
## Warning in ks.test.default(clean$SimpleReturn, sim_cauchy): p-value will be
## approximate in the presence of ties
```

```
print(cau)
```

```
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  clean$SimpleReturn and sim_cauchy
## D = 0.13508, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

```
results <- data.frame(Distribution = c("Normal", "Student's t", "Laplace",
    "Power Law", "Uniform", "Exponential", "Cauchy"), D_Statistic = c(nor$statistic,
    stu$statistic, lap$statistic, pow$statistic, uni$statistic, exp$statistic,
    cau$statistic))

print(results[order(results$D_Statistic), ])
```

### Test Results

```
##   Distribution D_Statistic
## 7       Cauchy   0.1350828
## 3      Laplace   0.2024805
## 1       Normal   0.2140647
## 2  Student's t   0.4945578
## 5      Uniform   0.6406632
## 6  Exponential   0.9815688
## 4    Power Law   0.9955866
```

The results of the Kolmogorov-Smirnov test show the D-statistic values for different distributions compared to the sample data. The Cauchy distribution has the smallest D value (0.135), indicating the closest fit, followed by the Laplace distribution (0.202). Other distributions, such as Normal (0.214), Student's t (0.495), and Exponential (0.632), show progressively larger D values, indicating a poorer fit. The Uniform (0.641) and Power Law (0.996) distributions have the highest D values, suggesting they are the least similar to the sample data.

**2. Test using Shapiro-Wilk normality test the Ethereum returns for trading data every five minutes, from August 7, 2015 to April 15, 2025.**

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.4.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 4.4.3
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
data = read.csv("ETHUSD_1m_Binance.csv")

data = data[data$Open.time <= as.POSIXct("2025-04-15 23:59:59", tz = "UTC"),
    ]

clean_data = data[, c("Open.time", "Close")]

clean_data$SimpleReturn <- c(NA, (diff(clean_data$Close)/head(clean_data$Close,
    -1)))

clean_data = na.omit(clean_data)

clean_data$Open.time <- as.POSIXct(clean_data$Open.time, format = "%Y-%m-%d %H:%M:%S",
    tz = "UTC")

# Group the 1 min interval data into 5 min

clean_data_5min <- clean_data %>%
    mutate(Time_5min = floor_date(Open.time, "5 minutes")) %>%
    group_by(Time_5min) %>%
    summarise(Open_5min = first(Close), Close_5min = last(Close), .groups = "drop") %>%
```

```
    mutate(SimpleReturn_5min = (Close_5min - Open_5min)/Open_5min) %>%
    select(Time_5min, SimpleReturn_5min) %>%
    filter(!is.na(SimpleReturn_5min))

summary(clean_data_5min$SimpleReturn_5min)
```

```
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -1.217e-01 -9.061e-04  0.000e+00 -6.880e-06  9.013e-04  1.211e-01
```

**Shapiro-Wilks Test** Importantly, we could not use the built-in function shapiro.test() since it is limited to data sets lower than 5000, while this particular data set is larger than 4 million.

```
manual_shapiro_wilk <- function(x) {
    x <- sort(x)
    n <- length(x)
    if (n < 3)
        stop("Sample size must be at least 3.")

    mean_x <- mean(x)
    ssd <- sum((x - mean_x)^2)

    m <- qnorm(((1:n) - 0.375)/(n + 0.25))
    m <- m - mean(m)

    a <- m/sqrt(sum(m^2))

    numerator <- (sum(a * x))^2

    W <- numerator/ssd

    if (n >= 3 & n <= 5000) {
        mu <- -1.2725 + 1.0521 * log(n)
        sigma <- 1.0308 - 0.26758 * log(n)
        y <- log(1 - W)
        z <- (y - mu)/sigma
        p_value <- pnorm(z)
    } else {
        mu <- -1.2725 + 1.0521 * log(5000)
        sigma <- 1.0308 - 0.26758 * log(5000)
        y <- log(1 - W)
        z <- (y - mu)/sigma
        p_value <- pnorm(z)
        warning("P-value approximation may not be reliable for sample sizes > 5000")
    }

    return(list(W = W, p_value = p_value))
}

result <- manual_shapiro_wilk(clean_data_5min$SimpleReturn_5min)
```

```
## Warning in manual_shapiro_wilk(clean_data_5min$SimpleReturn_5min): P-value
## approximation may not be reliable for sample sizes > 5000
```

```r
cat("\nManual Shapiro-Wilk W statistic:", result$W, "\n")
```

```
##
## Manual Shapiro-Wilk W statistic: 0.7610889
```

```r
cat("Approximate p-value:", result$p_value, "\n")
```

```
## Approximate p-value: 1
```

With a W statistic of 0.7610889, significantly lower than 1,typically indicating deviations from normality. However, the interpretation must consider the p-value.

An approximate p-value of 1 suggests that the data does not significantly deviate from a normal distribution. In statistical terms, this means that the null hypothesis (that the data is normally distributed) cannot be rejected.

With a dataset exceeding 4 million elements, the Shapiro-Wilk test is sensitive to even minor deviations from normality. However, the high p-value indicates that these deviations are not statistically significant.

The results imply that the dataset can be treated as normally distributed for the purposes of further statistical analysis. This allows for the use of parametric tests, which assume normality.

Github Link: https://github.com/SylTana/APM1110-QUIJANO-JULIAN_PHILIP/tree/main/SA2