Project I – Volatility and Signals in BTC

Ignacio C.

2025-12-22

Contents

1. Objective **2**

2. Libraries & Setup **2**

3. Data Download **2**

4. Returns & EDA **2**

5. Stationarity & Dependence **4**

6. GARCH(1,1) Model **6**

7. Model Diagnostics **9**

8. Rolling Forecast **11**

9. Volatility Regimes & Directional Signals 11

10. Signals Visualization 13

11. Interpretation & Model Assumptions **13**

1. Objective

Analyze BTC volatility and generate directional signals using GARCH(1,1), validating model assumptions and consistency of log returns.

2. Libraries & Setup

```r
library(quantmod)
library(xts)
library(tidyverse)
library(tseries)
library(rugarch)
library(moments)
library(zoo)
```

3. Data Download

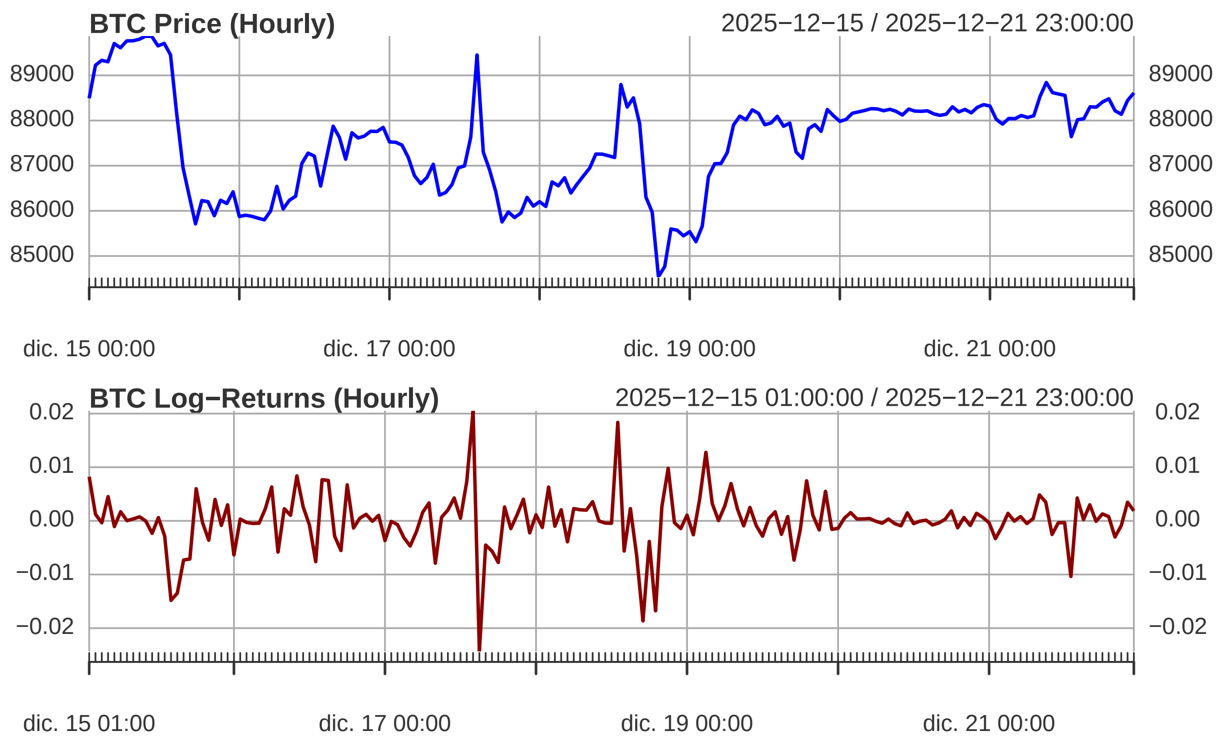
```r
# Download BTC hourly prices from Yahoo Finance
# NA values in closing prices are removed for consistency
getSymbols("BTC-USD", src = "yahoo", periodicity = "hourly")
```

```
## [1] "BTC-USD"
```

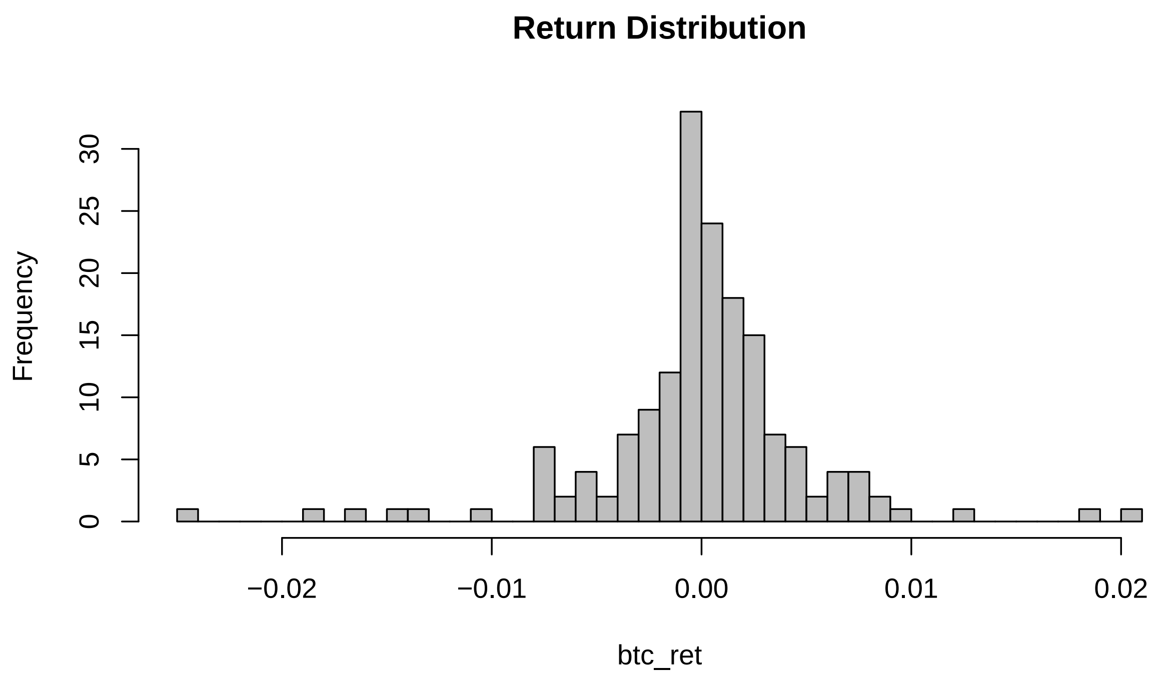
```r
btc <- `BTC-USD`
btc <- btc[!is.na(Cl(btc)), ]
btc_close <- Cl(btc)
```

4. Returns & EDA

```r
# Calculate log returns for BTC
# Provides stationary series for volatility modeling
# Plot price and returns to visually inspect trends and volatility
# Histogram and basic stats help check distribution shape, skewness, kurtosis
btc_ret <- diff(log(btc_close))
btc_ret <- na.omit(btc_ret)

# Plots
par(mfrow = c(2,1))
plot(btc_close, main = "BTC Price (Hourly)", col = "blue")
plot(btc_ret, main = "BTC Log-Returns (Hourly)", col = "darkred")
```



```r
par(mfrow = c(1,1))

# Histogram + Statistics
hist(btc_ret, breaks = 50, col = "gray", main = "Return Distribution")
```



```r
mean(btc_ret)
```

```
## [1] 7.983133e-06
```

```r
sd(btc_ret)
```

```
## [1] 0.005172996
```

```r
kurtosis(btc_ret)
```

```
## BTC-USD.Close
##      8.761379
```

```r
skewness(btc_ret)
```

```
## BTC-USD.Close
##    -0.6006295
```

5. Stationarity & Dependence

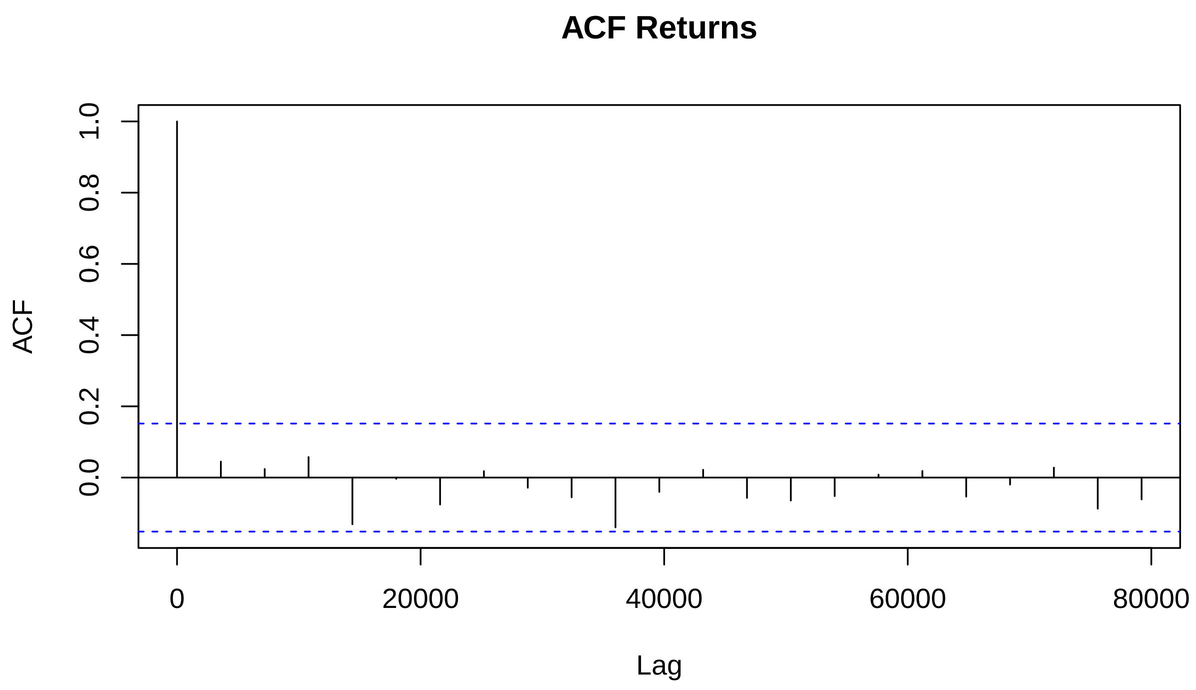
```r
adf.test(btc_ret)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  btc_ret
## Dickey-Fuller = -5.8586, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

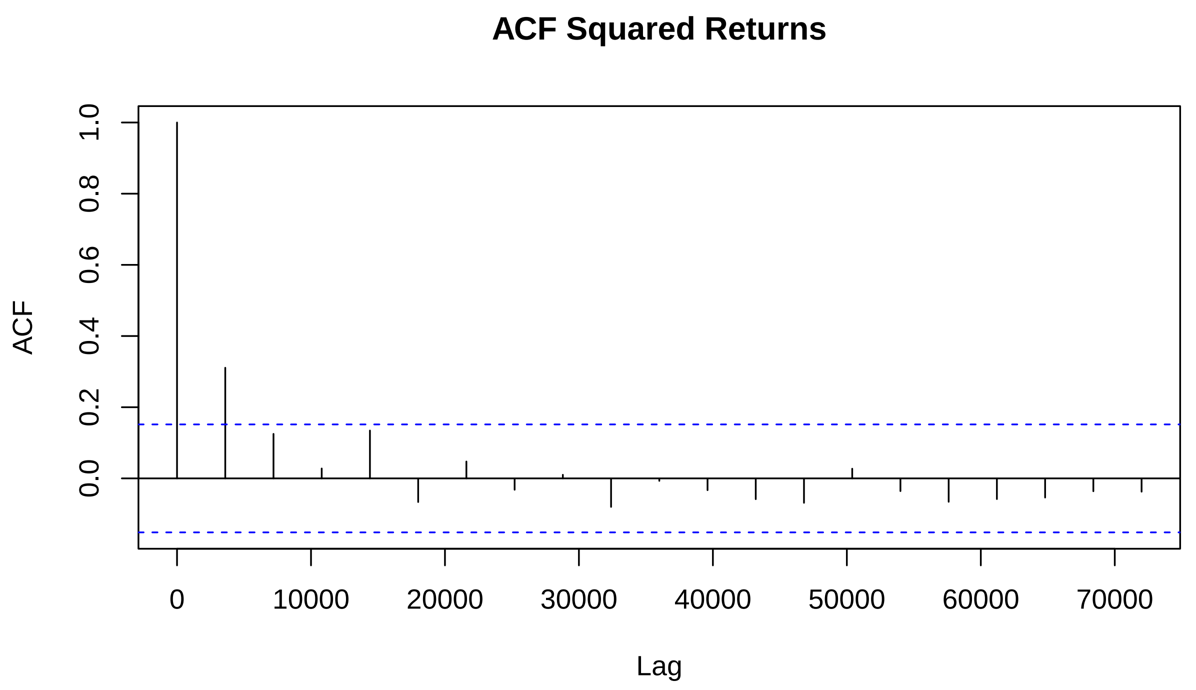
```r
kpss.test(btc_ret)
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  btc_ret
## KPSS Level = 0.063898, Truncation lag parameter = 4, p-value = 0.1
```

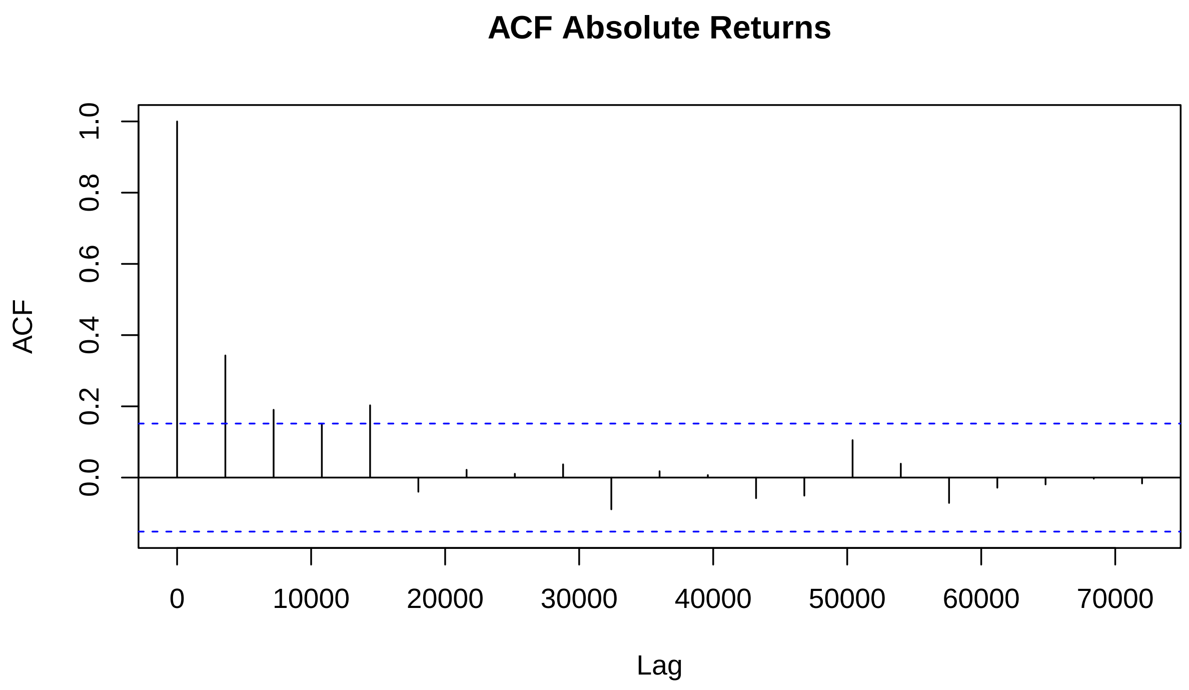
```r
acf(btc_ret, main = "ACF Returns")
```



```
acf(btc_ret^2, lag.max = 20, main = "ACF Squared Returns")
```



```
acf(abs(btc_ret), lag.max = 20, main = "ACF Absolute Returns")
```



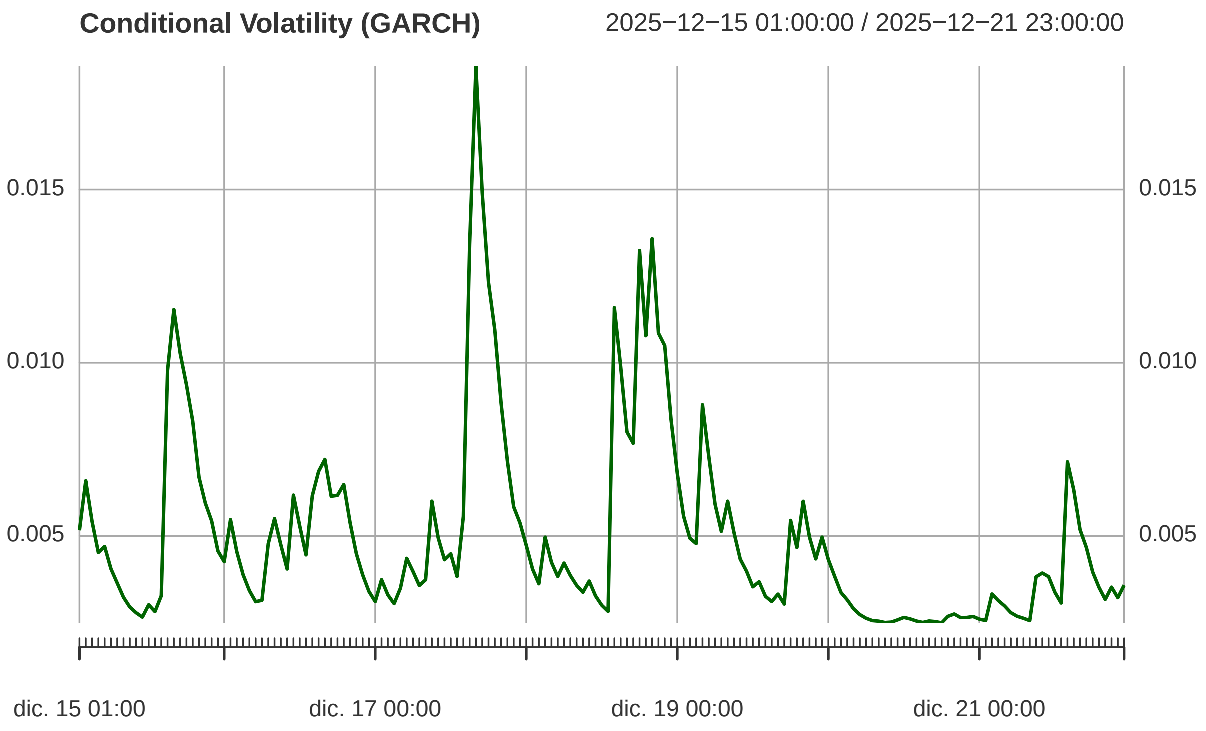
6. GARCH(1,1) Model

```r
# Specify and fit a standard GARCH(1,1) model with t-distribution
# Captures time-varying conditional volatility in returns
# sigma(fit) gives estimated volatility over time

spec <- ugarchspec(
  variance.model = list(model = "sGARCH", garchOrder = c(1,1)),
  mean.model = list(armaOrder = c(0,0), include.mean = TRUE),
  distribution.model = "std"
)

fit <- ugarchfit(spec = spec, data = btc_ret)
## GARCH Model Fit - Focus on alpha1, beta1, shape, model fit
show(fit)
```

```
##
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
##
## Conditional Variance Dynamics
## -----------------------------------
## GARCH Model  : sGARCH(1,1)
## Mean Model   : ARFIMA(0,0,0)
## Distribution : std
```

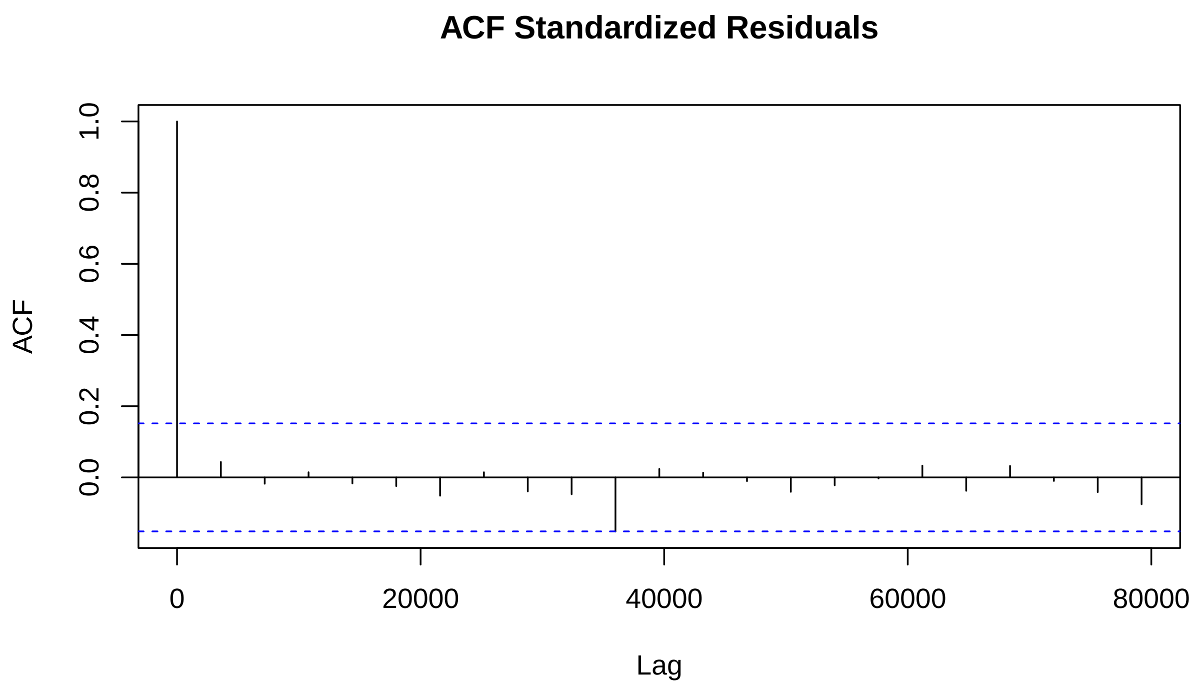
```
##
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      0.000196    0.000145  1.35274 0.176139
## omega   0.000002    0.000011  0.21523 0.829587
## alpha1  0.385069    0.145782  2.64140 0.008256
## beta1   0.613931    0.330946  1.85508 0.063585
## shape   2.732173    0.211139 12.94014 0.000000
##
## Robust Standard Errors:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      0.000196    0.001422 0.138107  0.89016
## omega   0.000002    0.000150 0.015231  0.98785
## alpha1  0.385069    0.530117 0.726384  0.46760
## beta1   0.613931    4.908440 0.125077  0.90046
## shape   2.732173    5.882533 0.464455  0.64232
##
## LogLikelihood : 693.215
##
## Information Criteria
## ------------------------------------
##
## Akaike       -8.2421
## Bayes        -8.1487
## Shibata      -8.2438
## Hannan-Quinn -8.2042
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                     0.3213  0.5708
## Lag[2*(p+q)+(p+q)-1][2]    0.3484  0.7693
## Lag[4*(p+q)+(p+q)-1][5]    0.4268  0.9684
## d.o.f=0
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                  0.0001806  0.9893
## Lag[2*(p+q)+(p+q)-1][5] 0.8030491  0.9024
## Lag[4*(p+q)+(p+q)-1][9] 1.3534821  0.9667
## d.o.f=2
##
## Weighted ARCH LM Tests
## ------------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[3]    0.1527 0.500 2.000  0.6960
## ARCH Lag[5]    0.9291 1.440 1.667  0.7541
## ARCH Lag[7]    1.1173 2.315 1.543  0.8938
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  1.3807
## Individual Statistics:
## mu     0.0430
## omega  0.2889
## alpha1 0.6747
## beta1  0.3806
## shape  0.4896
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:          1.28 1.47 1.88
## Individual Statistic:     0.35 0.47 0.75
##
## Sign Bias Test
## ------------------------------------
##                    t-value   prob sig
## Sign Bias           1.1902 0.2357
## Negative Sign Bias  0.8045 0.4223
## Positive Sign Bias  0.5023 0.6161
## Joint Effect        1.5330 0.6747
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20     18.99       0.4576
## 2    30     37.25       0.1399
## 3    40     41.62       0.3573
## 4    50     40.49       0.8016
##
##
## Elapsed time : 0.06182289
```

```
# Conditional volatility plot
plot(sigma(fit), main = "Conditional Volatility (GARCH)", col = "darkgreen")
```

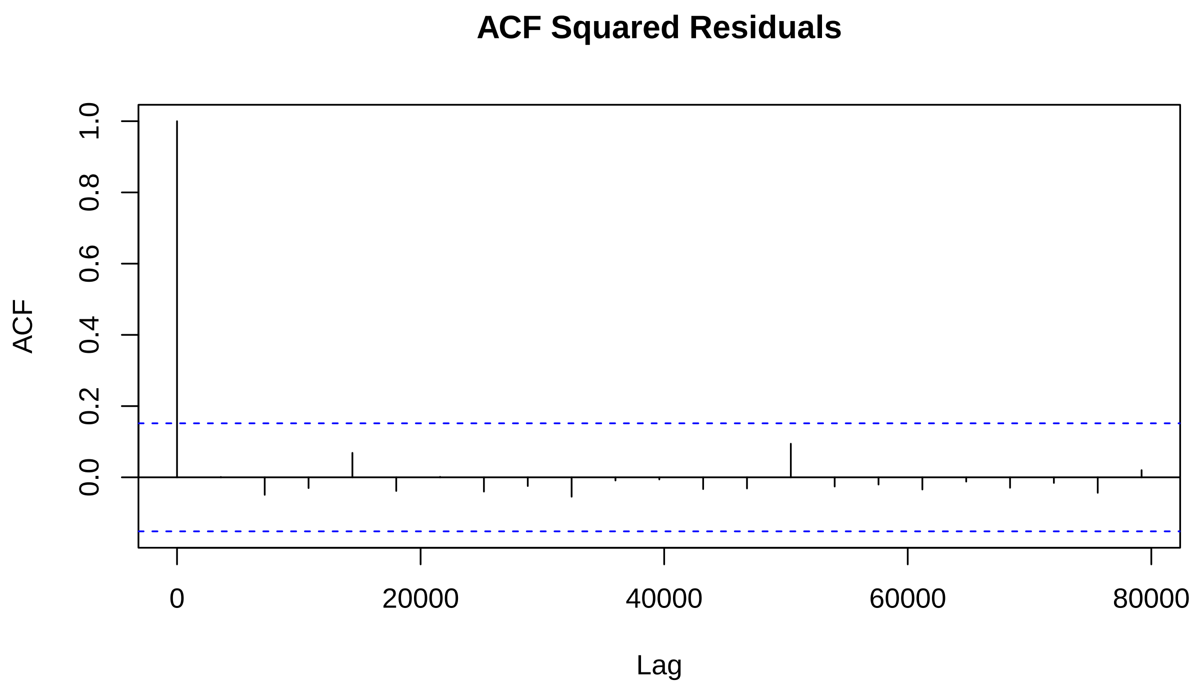


7. Model Diagnostics

```r
# Examine standardized residuals for autocorrelation
# Squared residuals ACF checks for remaining volatility clustering
# Box-Ljung test assesses if residuals are approximately independent
resid <- residuals(fit, standardize = TRUE)

acf(resid, main = "ACF Standardized Residuals")
```



```
acf(resid^2, main = "ACF Squared Residuals")
```



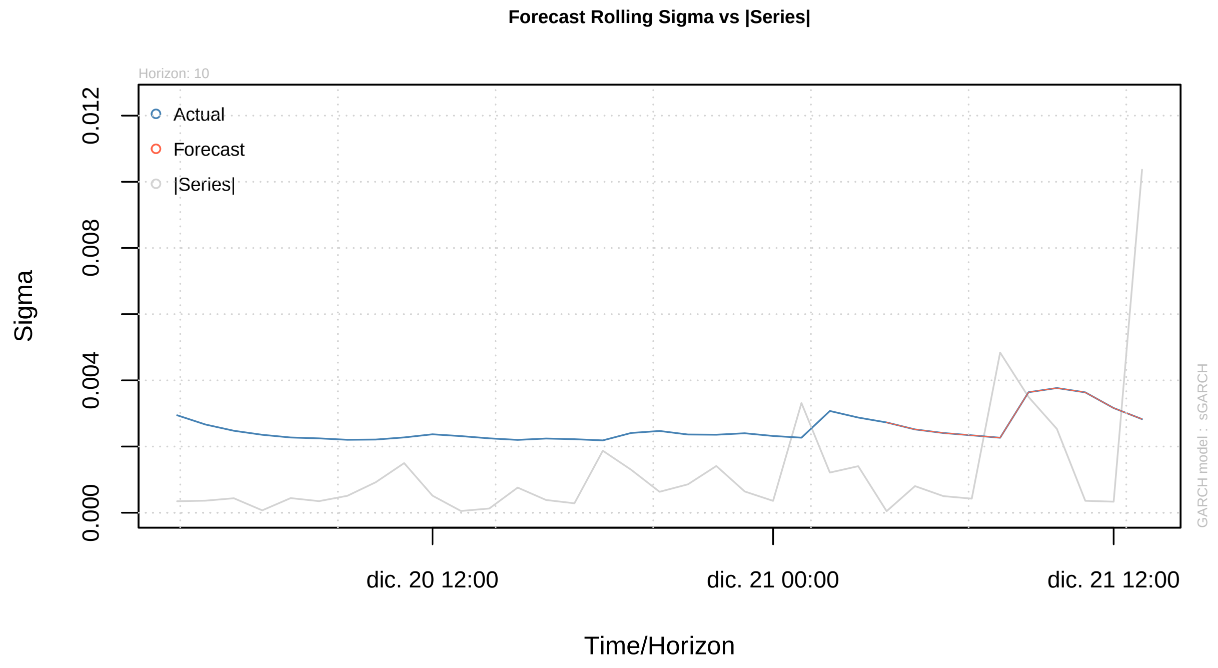
```
Box.test(resid^2, lag = 20, type = "Ljung")
```

```
##
```

```
##  Box-Ljung test
##
## data:  resid^2
## X-squared = 5.2659, df = 20, p-value = 0.9996
```

8. Rolling Forecast

```
# Generate out-of-sample rolling forecasts of volatility
# Useful to see expected volatility changes in near future
# n.roll parameter mimics a moving forecast horizon
# Rolling forecast setup
fit_roll <- ugarchfit(spec = spec, data = btc_ret, out.sample = 20)
forecast_vol <- ugarchforecast(fitORspec = fit_roll, n.ahead = 10, n.roll = 10)
plot(forecast_vol, which = 4)
```



9. Volatility Regimes & Directional Signals

This section and *.10* illustrates a **practical application** of the GARCH model and logit-based directional predictions. We identify "calm" market periods (volatility below the 40th percentile) and estimate the probability of an upward move in BTC for the next period. Signals are classified conservatively as LONG, SHORT, or NO TRADE based on predicted probabilities.

> **Note:** This is an illustrative exercise to show how the model outputs can be interpreted. It is **not a trading strategy** or a backtest.

```r
sigma_hat <- sigma(fit)
sigma_thresh <- quantile(sigma_hat, 0.4, na.rm = TRUE)
calm_vec <- sigma_hat < sigma_thresh

ret_vec <- as.numeric(coredata(btc_ret))
trend_6h <- rollapply(btc_ret, width = 6, FUN = sum, align = "right", fill = NA)
trend6h_vec <- as.numeric(coredata(trend_6h))
direction_vec <- ifelse(as.numeric(coredata(dplyr::lead(btc_ret, n = 1))) > 0, 1, 0)

data_dir <- data.frame(
  ret = ret_vec,
  trend6h = trend6h_vec,
  sigma = as.numeric(coredata(sigma_hat)),
  direction = direction_vec,
  calm = calm_vec
)
data_dir <- na.omit(data_dir)
data_calm <- subset(data_dir, calm == TRUE)

model_logit <- glm(direction ~ trend6h, data = data_calm, family = binomial)
summary(model_logit)
```

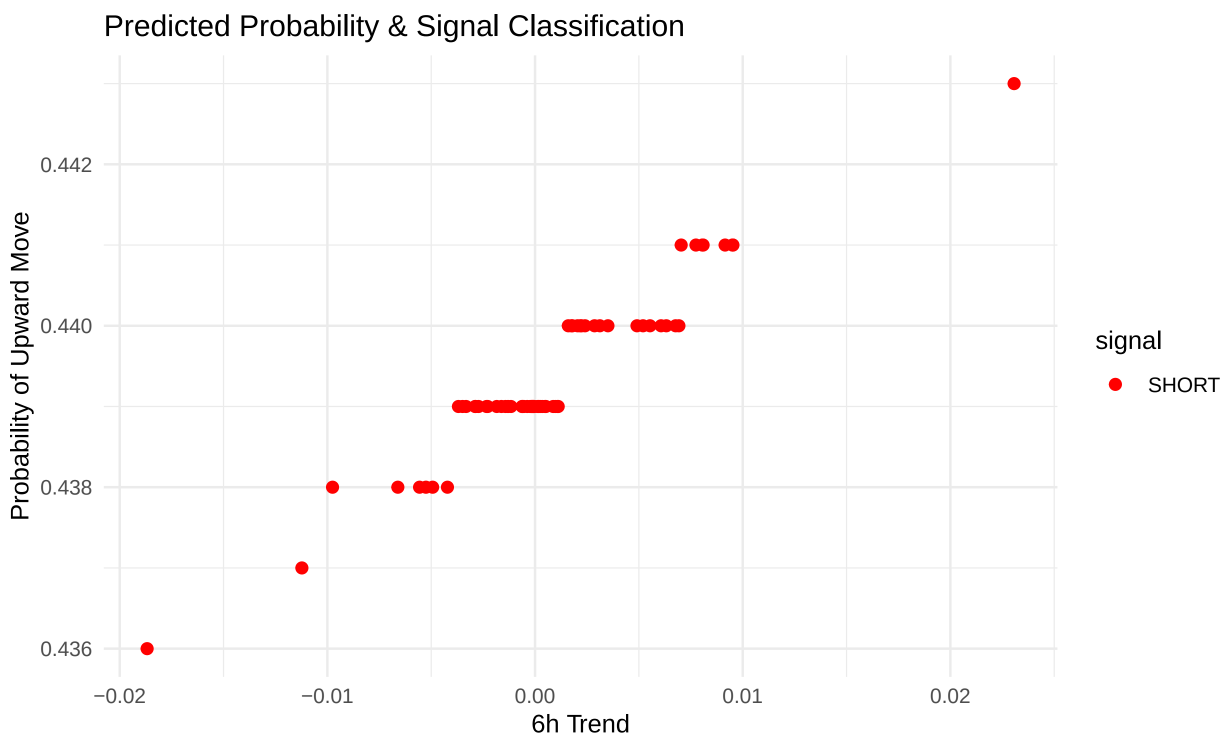
```
##
## Call:
## glm(formula = direction ~ trend6h, family = binomial, data = data_calm)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.2442     0.2505  -0.975    0.330
## trend6h       0.7245    43.8772   0.017    0.987
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 90.523  on 65  degrees of freedom
## Residual deviance: 90.523  on 64  degrees of freedom
## AIC: 94.523
##
## Number of Fisher Scoring iterations: 3
```

```r
prob_up <- predict(model_logit, type = "response")
signal <- ifelse(prob_up > 0.55, "LONG",
                 ifelse(prob_up < 0.45, "SHORT", "NO TRADE"))

signals_table <- data.frame(
  trend6h = data_calm$trend6h,
  prob_up = round(prob_up, 3),
  signal = signal,
  confidence = abs(prob_up - 0.5)
)

head(signals_table, 10)
```

```
##                             trend6h prob_up signal confidence
## 2025-12-15 07:00:00  0.0060674391   0.440  SHORT 0.05966745
## 2025-12-15 08:00:00  0.0051996572   0.440  SHORT 0.05982240
## 2025-12-15 09:00:00  0.0063210138   0.440  SHORT 0.05962218
## 2025-12-15 10:00:00  0.0017521199   0.440  SHORT 0.06043784
## 2025-12-15 11:00:00  0.0004816593   0.439  SHORT 0.06066459
## 2025-12-15 12:00:00 -0.0006103030   0.439  SHORT 0.06085947
## 2025-12-15 13:00:00 -0.0034926796   0.439  SHORT 0.06137377
## 2025-12-15 14:00:00 -0.0186796067   0.436  SHORT 0.06408138
## 2025-12-16 04:00:00 -0.0042178260   0.438  SHORT 0.06150314
## 2025-12-16 05:00:00 -0.0049321215   0.438  SHORT 0.06163056
```

10. Signals Visualization

```r
ggplot(signals_table, aes(x = trend6h, y = prob_up, color = signal)) +
  geom_point(size = 2) +
  scale_color_manual(values = c("LONG" = "green", "SHORT" = "red", "NO TRADE" = "gray")) +
  labs(title = "Predicted Probability & Signal Classification",
       x = "6h Trend",
       y = "Probability of Upward Move") +
  theme_minimal()
```



11. Interpretation & Model Assumptions

- Conditional volatility modeled by GARCH(1,1)
- Standardized residuals approximately independent

- Logistic regression predicts next-period directional move
- Calm periods = volatility below 40th percentile
- LONG if prob >0.55, SHORT if <0.45, else NO TRADE
- Confidence = distance from 0.5 probability