```
set.seed(123)
library(stochvol)
library(tseries)
# Price Data
market_price <- read_csv("C:/Anirudh/M.Sc. Semester IV/Financial Econometrics I/Term Paper/market-price.csv")
View(market_price)
colnames(market_price) <- c('Date','Price')
prices <- market_price$Price
logprices <- log(prices)
priceplot <- data.frame(as.Date(market_price$Date), market_price$Price)
plot(priceplot, type = 'l', col = 'blue', lwd = '2', ylab = "", xlab = "",
     main = 'Historical Bitcoin Prices ($)')

# Volume Data
trade_volume <- read_csv("C:/Anirudh/M.Sc. Semester IV/Financial Econometrics I/Term Paper/trade-volume.csv")
View(trade_volume)
colnames(trade_volume) <- c('Date','Volume')
plot(trade_volume$Volume, type = 'l')
volumes <- trade_volume$Volume

# Returns
returns <- logprices[2:length(logprices)] - logprices[1:(length(logprices)-1)]
plot(returns, type = 'l')
plot(prices, type = 'l')
plot(logprices, type = 'l')



#########################################################
#########################################################
# BITCOIN ###############################################
#########################################################
bitcoin_volume <- trade_volume[547:nrow(trade_volume),]
bitcoin_volume$Date <- as.Date(bitcoin_volume$Date)
plot(bitcoin_volume, type = 'l', main = 'Volume Traded', col = 'blue')

bitcoin_prices <- market_price[547:nrow(market_price),]
bitcoin_prices$Date <- as.Date(bitcoin_prices$Date)
View(bitcoin_prices)
plot(bitcoin_prices, type = 'l')
bitcoin_ret <- logret(bitcoin_prices$Price, demean = TRUE)
plot(bitcoin_prices$Date[2:nrow(bitcoin_prices)], bitcoin_ret, type = 'l',
     col = 'purple', main = 'Bitcoin Returns', ylab = "", lwd = '2')
plot(bitcoin_prices$Date[2:nrow(bitcoin_prices)], sqrt(bitcoin_ret^2), type = 'l',
     col = 'brown', main = 'Bitcoin Observed Volatility', ylab = "", lwd = '2')
adf.test(bitcoin_ret)$p.value
PP.test(bitcoin_ret)
#########################################################
# Stochastic Volatility Estimates
res <- svsample(bitcoin_ret, priormu = c(-10, 1), priorphi = c(20,1.5),priorsigma = 0.1)
summary(res, showlatent = TRUE)
mean_exp_ht_2 <- as.numeric(summary(res, showlatent = TRUE)$latent[,6])
sd_exp_ht_2 <- as.numeric(summary(res, showlatent = TRUE)$latent[,7])
volatility <- sd_exp_ht_2
stoch_vol <- data.frame(bitcoin_prices$Date, volatility)
colnames(stoch_vol) <- c('Date','volatility')
stoch_vol$volatility <- stoch_vol$vol
stoch_vol <- stoch_vol[2:nrow(stoch_vol),]
plot(stoch_vol, type = 'l', main = 'Estimated (Median) Stochastic Volatility', col = 'brown',
     lwd = '2')
volplot(res,dates = bitcoin_prices$Date[-1])
plot(res, dates = bitcoin_prices$Date[-1])
plot(res, forecast = 10, dates = bitcoin_prices$Date[-1])
plot(resid(res),bitcoin_ret)

RMSE_stochvol <- sqrt((1/nrow(stoch_vol))*sum(stoch_vol[,2]^2 - (sqrt(bitcoin_ret^2))^2)^2)
print(RMSE_stochvol)
summary(res, showlatent = FALSE)

#########################################################

# GARCH(1,1) Volatility Estimates using library fGarch
bitcoin_returns_zoo <- zoo(bitcoin_ret,order.by = as.Date(stoch_vol$Date))
library(fGarch)
fit = garchFit( ~ garch(1, 1), data = bitcoin_returns_zoo)
garchEstimates <- volatility(fit, type = 'sigma')
predict(fit,1)
plot(bitcoin_prices$Date[-1], garchEstimates, type = 'l',
     col = 'brown', lwd = "2", main = 'Estimated GARCH(1,1) Volatility')
#########################################################

# Bitcoin Volatility vs GARCH volatility vs Observed Volatility
plot(stoch_vol, type = 'l', col = 'black', lwd = "2", ylim = range(0,1),
     main = 'Comparing Volatility Estimates')
par(new = TRUE)
```

```
plot(garchEstimates, type = 'l', col = 'red', lwd = "2",
     ylim = range(0,1), axes = FALSE, xlab = "", ylab = "")
par(new = TRUE)
plot(sqrt(bitcoin_ret^2), type = 'l', col = 'blue', lwd = "1",
     ylim = range(0,1), axes = FALSE, xlab = "", ylab = "")
legend('topright', c('SV', 'GARCH','Observed'), lty = c(1,1,1),
       lwd = c(2,2,1), col = c('black', 'red', 'blue'))
#########################################################

# Rolling Window GARCH Volatility Estimation
w <- 100 # window size set to ~ 10% of the data size
rollingWindowGarch <- garchEstimates[1:w]
fit_w <- garchFit( ~ garch(1, 1), data = bitcoin_returns_zoo[1:w])
for(i in (w+1):length(garchEstimates)){
  fit_w <- garchFit( ~ garch(1, 1), data = bitcoin_returns_zoo[(i-(w-1)):i])
  rollingWindowGarch <- append(rollingWindowGarch,
                               as.numeric(predict(fit_w,1)[3]))
}

print(rollingWindowGarch)
plot(bitcoin_prices$Date[-1], garchEstimates, type = 'l',
     col = 'brown', lwd = "2",
     main = 'Comparing Rolling Window and Plain Vanilla GARCH Volatilities')
par(new = TRUE)
plot(rollingWindowGarch, type = 'l',
     axes = FALSE, xlab = "", ylab = "",
     col = 'black', lty = 2, lwd = 2)
legend('topright', c('GARCH', 'RW-GARCH'), lty = c(1,2),
       lwd = c(2,2), col = c('brown', 'black'))
#########################################################

# RMSE of the 3 Models
RMSE_garch <- sqrt((1/nrow(stoch_vol))*sum(garchEstimates^2 - (sqrt(bitcoin_ret^2))^2)^2)
RMSE_stochvol <- sqrt((1/nrow(stoch_vol))*sum(stoch_vol[,2]^2 - (sqrt(bitcoin_ret^2))^2)^2)
RMSE_RWgarch <- sqrt((1/nrow(stoch_vol))*sum(rollingWindowGarch^2 - (sqrt(bitcoin_ret^2))^2)^2)
print(RMSE_garch)
print(RMSE_stochvol)
print(RMSE_RWgarch)
#########################################################
```