1. a) $B(t) \sim N(0, t)$, $t_{n/d} = 1$

$$E\left(\sum_{i=1}^{n/d} \left(B(t_i) - B(t_{i-1}) \right)^2 \right) \xrightarrow{P} 1 .$$

So it doesn't depend on d.

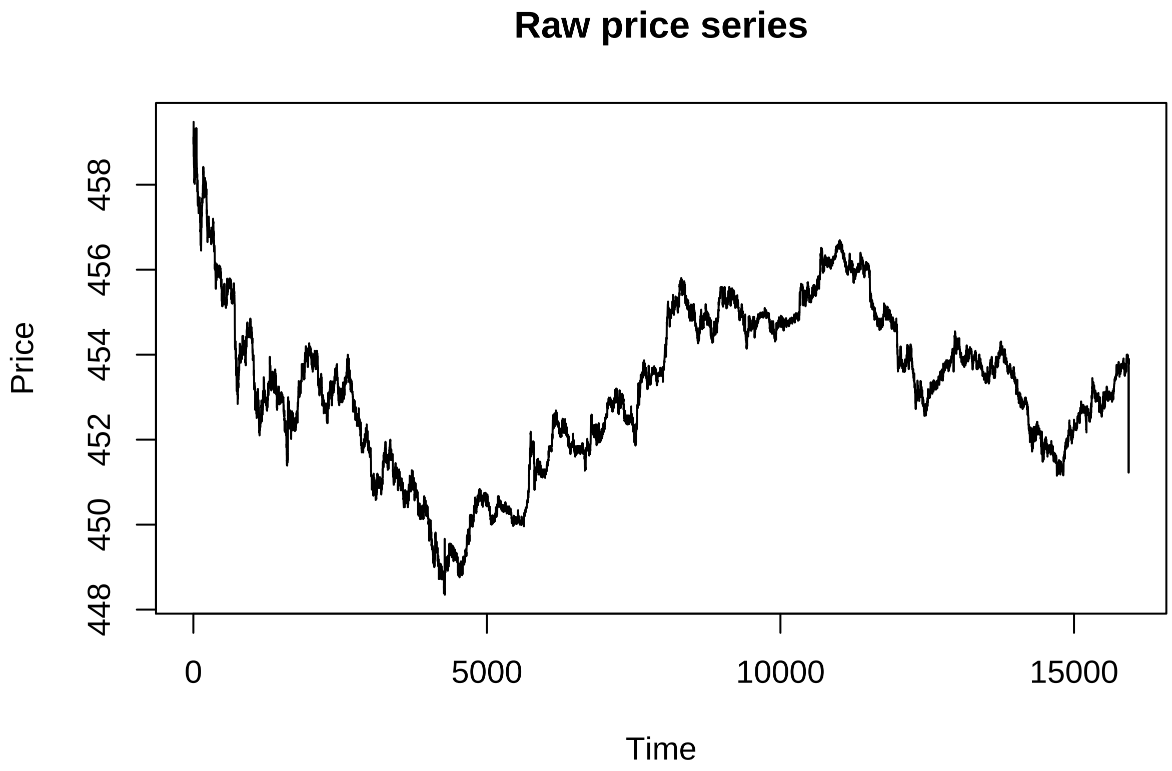
b) Due to the iid standard normal variable, ε_i, as d increases, the noise associated from ε_i also increases.

EX3.R

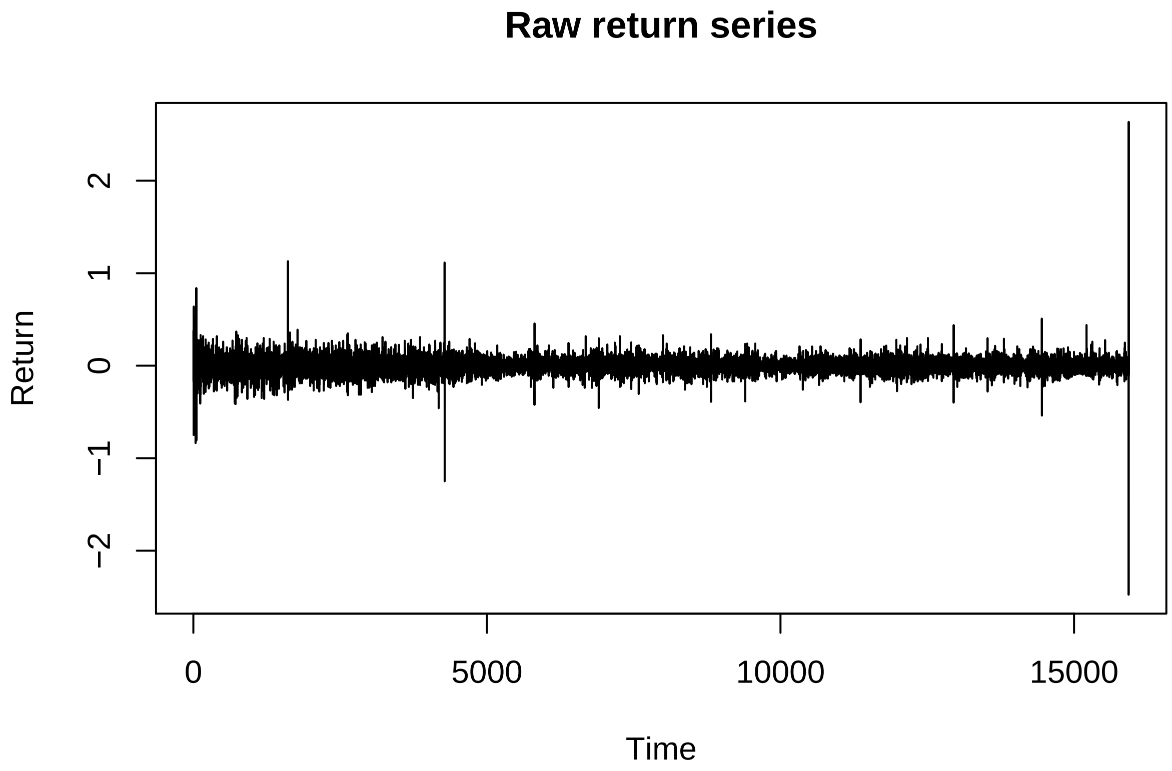
joonkang

2021-02-10

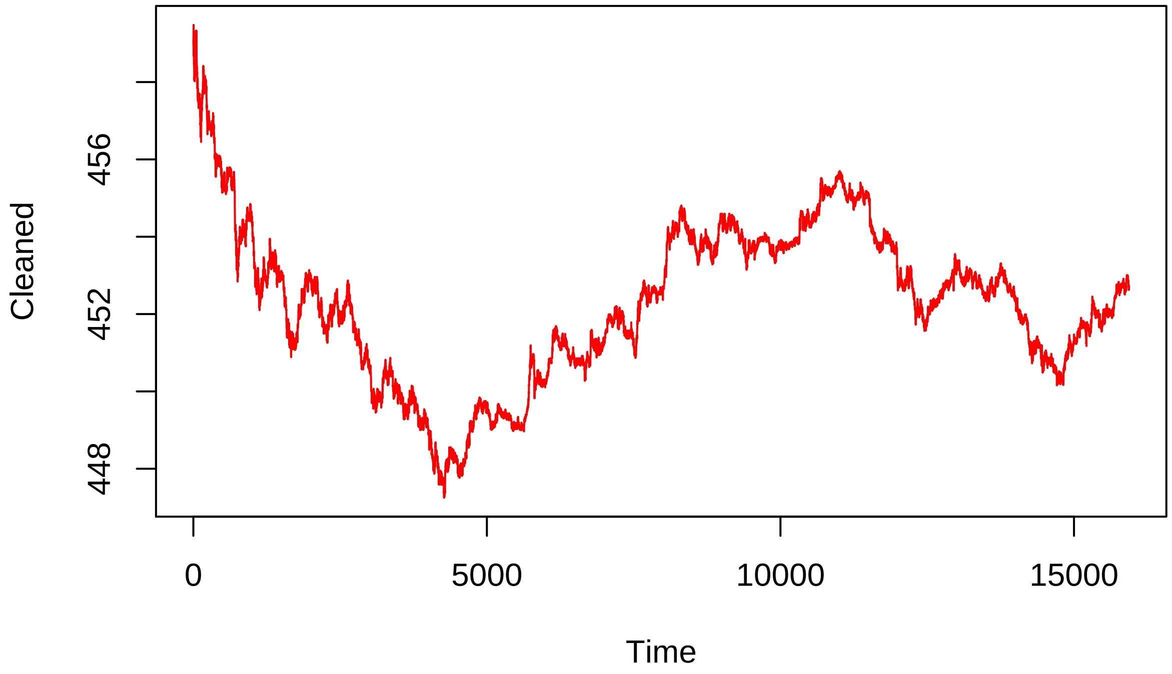
```r
#Q2.a
df <- read.csv("https://raw.githubusercontent.com/Anko-Jipsa/statistics/master/ST436/EX3/ysvmqmcmqg0wcbr

# Sampling the price series by each second.
sample_sec <- function(df) {
  tme = strptime(as.character(df[["TIME"]]), "%H:%M:%S")
  df_sec = df[as.logical(diff(tme$sec)), ]
  return(df_sec)
}

df_sec = sample_sec(df)
raw_p = df_sec[["PRICE"]]

#Q2.b
# There are no notable outliers based on reviewing the price series except the last trading minutes.
ts.plot(raw_p, ylab="Price", main="Raw price series")
```



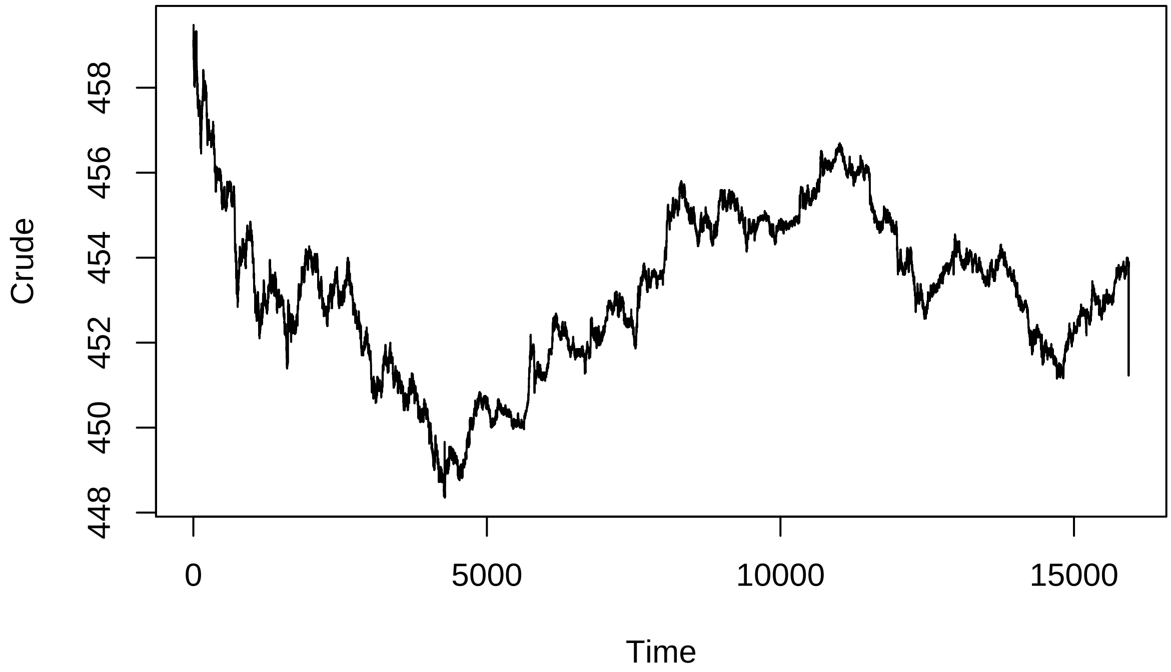
```r
ts.plot(diff(raw_p, lag=1), ylab="Return", main="Raw return series")
```



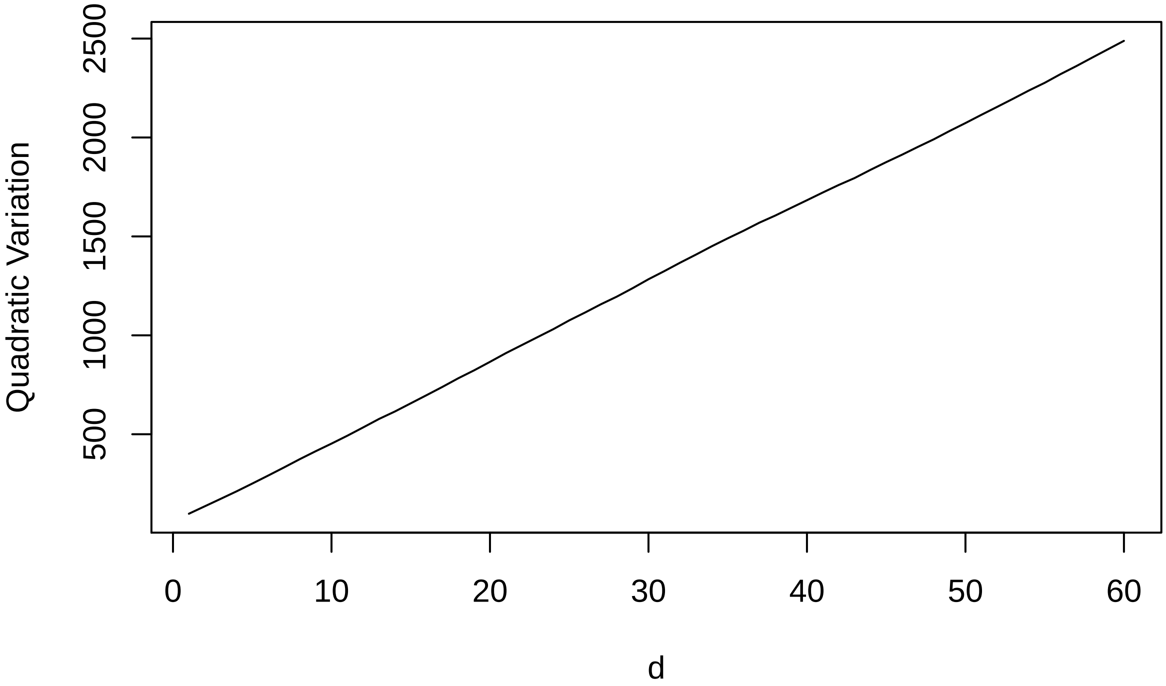
```
# Function to clean the price series given threshold.
price_filter = function(p_series, thresh=1.0){
  ret = diff(p_series)
  ret[abs(ret) > thresh] = 0
  output = cumsum(c(p_series[1], ret))
  return(output)
}

filtered_p = price_filter(raw_p)

# The outliers observed in the last trading minutes are removed.
ts.plot(filtered_p, col="red", ylab="Cleaned")
```



```r
ts.plot(raw_p, ylab="Crude")
```



```
#Q2.c
quad_var = function(p_series, diff_Sec){
  sse = sum(diff(filtered_p, lag=diff_Sec)^2)
  return(sse)
}

n_series = seq(1, 60, by=1)
quad_var_series = rep(0, length(n_series))
for (i in n_series){
  quad_var_series[i] = quad_var(filtered_p, i)
}

# The chart represents there is no microstructure noise.
plot(quad_var_series, type="l", xlab="d", ylab="Quadratic Variation")
```


Quadratic Variation
500
1000
1500
2000
2500
0
10
20
30
40
50
60
d