FE570 - Project Empirical analysis of microstructure data.

Naveen,Vineel,Amod,Amin

2024-11-24

Empirical analysis of microstructure data.

1. Perform a study of liquidity: compute the spread measures (quoted spread, effective spread, realized spread) in time buckets and study the intra-day liquidity dynamics

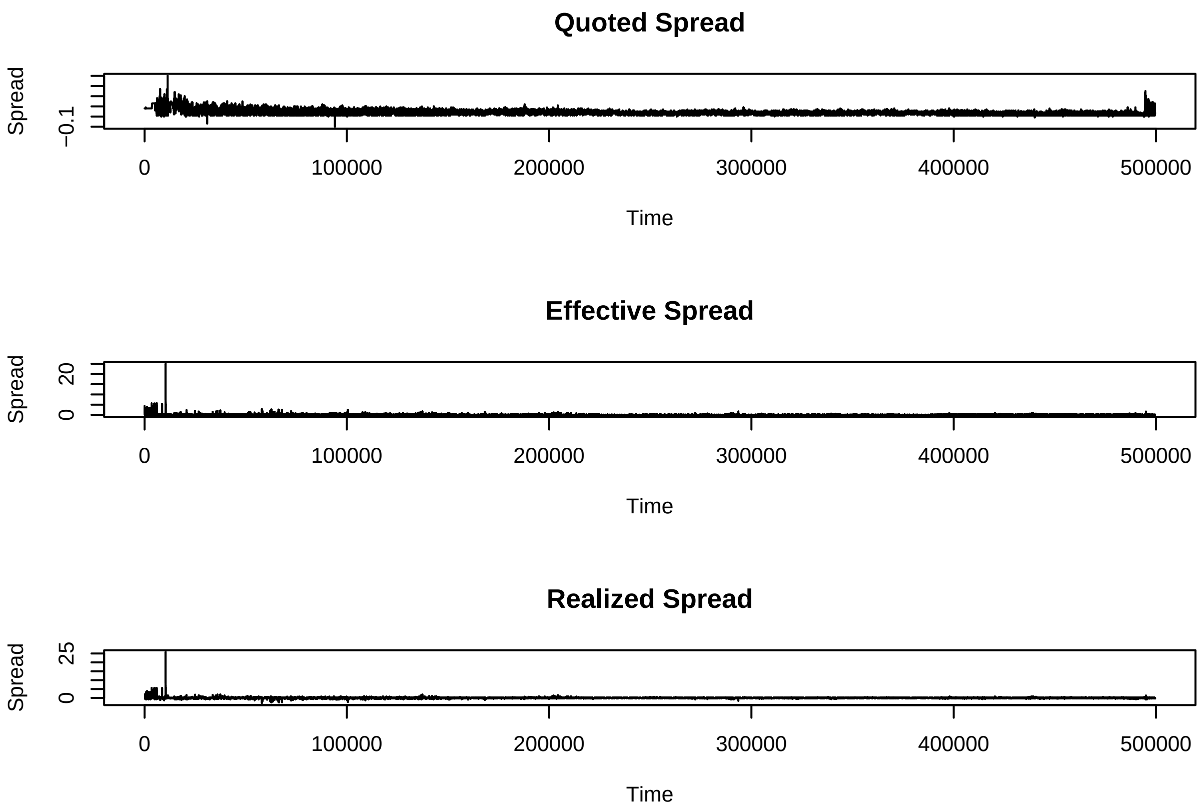
```
tsla.taq<-read.csv("tsla_taq_20241021_condensed.csv")

tsla.taq<- tsla.taq%>%select(Date.Time,Ex.Cntrb.ID,Bid.Price,Bid.Size,Ask.Price,Ask.Size,Tick.Dir.,Pric

tsla.taq$Date.Time<-as.nanotime(tsla.taq$Date.Time)
#trades$Date.Time<-ymd_hms(trades$Date.Time)
tsla.taq$Ex.Cntrb.ID<-as.factor(tsla.taq$Ex.Cntrb.ID)
tsla.taq$Tick.Dir.<-as.factor(tsla.taq$Tick.Dir.)
tsla.taq <- as.data.table(tsla.taq)


tsla.taq<-tsla.taq%>%mutate(DT=Date.Time,Exchange=Ex.Cntrb.ID,BID=Bid.Price,OFR=Ask.Price,BID.SIZE=Bid.S
```

```
tsla.taq.adf<-tsla.taq%>%filter(Exchange=="ADF")
#tsla.taq.adf<-trades%>%filter(Exchange=="NAS")

# Calculate effective spread
quoted_spread <- tsla.taq.adf$OFR - tsla.taq.adf$BID
mean(quoted_spread)
```

```
## [1] 0.04254817
```

```
# Calculate midpoint
midpoint <- (tsla.taq.adf$OFR + tsla.taq.adf$BID) / 2

# Calculate effective spread
effective_spread <- 2 * abs(tsla.taq.adf$PRICE - midpoint)

# Set time horizon for future midpoint
future_interval <- 5 * 60  # 5 minutes in seconds

# Shift midpoint by future_interval to estimate future midpoint
future_midpoint <- shift(midpoint, future_interval)
```

```
# Calculate realized spread
realized_spread <- 2 * (tsla.taq.adf$PRICE - future_midpoint)

# Plotting
par(mfrow = c(3, 1))  # Split plots
plot(quoted_spread, type = "l", main = "Quoted Spread", ylab = "Spread", xlab = "Time")
plot(effective_spread, type = "l", main = "Effective Spread", ylab = "Spread", xlab = "Time")
plot(realized_spread, type = "l", main = "Realized Spread", ylab = "Spread", xlab = "Time")
```



2. Estimate the volatility using intraday data

```
# loads a xts file called tqdataMktHrs
tqdata <- tsla.taq.adf

head(tqdata)
```

```
##                                     DT Exchange    BID BID.SIZE    OFR OFR.SIZE
## 1: 2024-10-21T12:00:00.033046979+00:00      ADF 217.85       13 217.93        5
## 2: 2024-10-21T12:00:00.033326923+00:00      ADF 217.85       13 217.93        5
## 3: 2024-10-21T12:00:00.033326923+00:00      ADF 217.85       13 217.93        5
## 4: 2024-10-21T12:00:00.053015660+00:00      ADF 217.85       13 217.93        5
## 5: 2024-10-21T12:00:00.058260942+00:00      ADF 217.85       13 217.93        5
## 6: 2024-10-21T12:00:00.058260942+00:00      ADF 217.85       13 217.93        5
##    Tick.Dir.    PRICE SIZE
## 1:           220.0817    1
```

```
## 2:           220.0817    1
## 3:           220.0817    1
## 4:           220.0817    1
## 5:           219.5357    1
## 6:           219.5357    1
```

```r
tail(tqdata)
```

```
##                                      DT Exchange   BID BID.SIZE    OFR OFR.SIZE
## 1: 2024-10-21T23:59:52.798013043+00:00      ADF 218.5        4 218.55       30
## 2: 2024-10-21T23:59:52.948029577+00:00      ADF 218.5        4 218.55       30
## 3: 2024-10-21T23:59:54.058056449+00:00      ADF 218.5        4 218.55       30
## 4: 2024-10-21T23:59:56.043247510+00:00      ADF 218.5        4 218.55       30
## 5: 2024-10-21T23:59:56.078043716+00:00      ADF 218.5        4 218.55       30
## 6: 2024-10-21T23:59:57.948011717+00:00      ADF 218.5        4 218.55       30
##    Tick.Dir.    PRICE SIZE
## 1:           218.5000  100
## 2:           218.5250    1
## 3:           218.5007   10
## 4:           218.5000   25
## 5:           218.5000    1
## 6:           218.5000  100
```

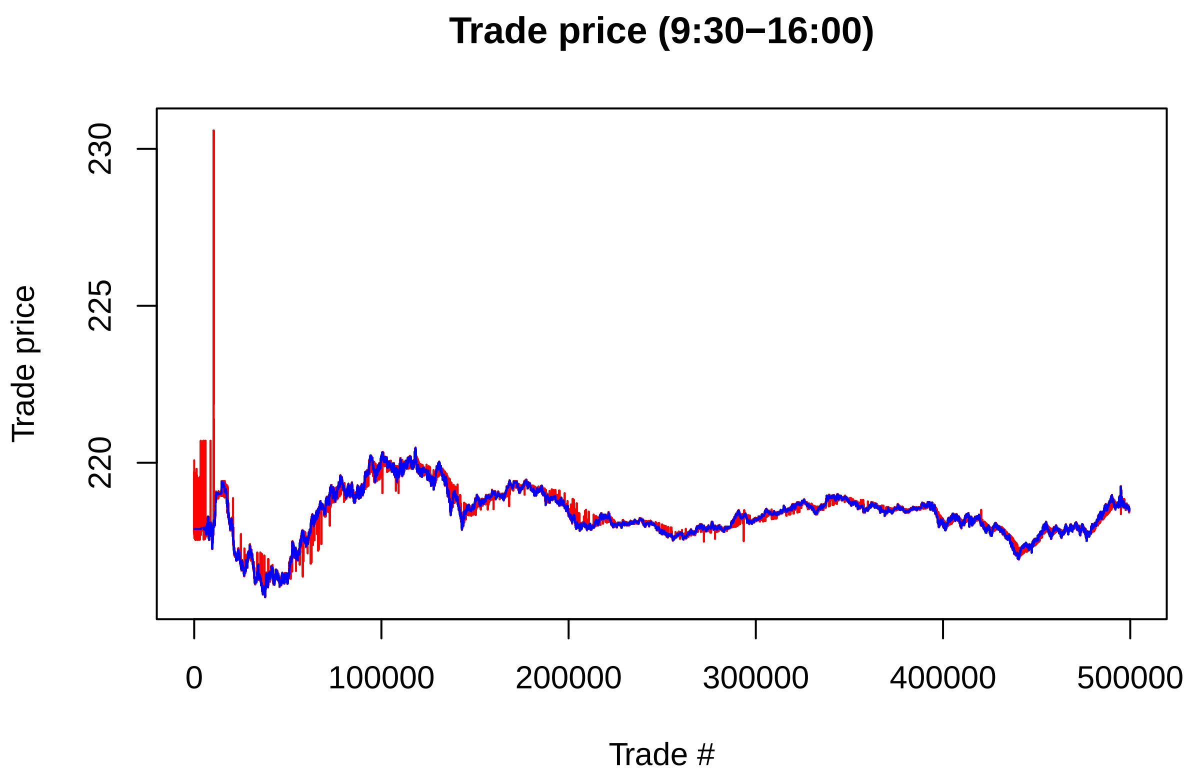
```r
length(tqdata$SIZE)
```

```
## [1] 499527
```

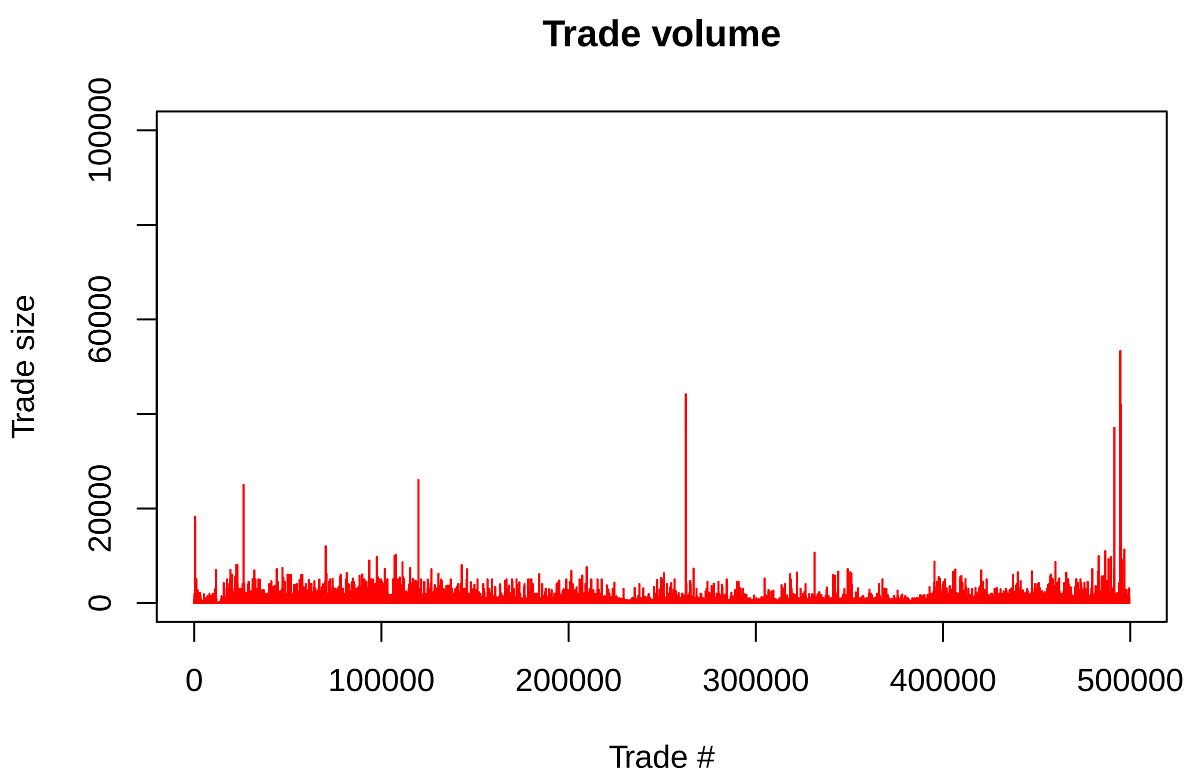
```r
# summarize trades by exchange

length(tqdata$SIZE)
```

```
## [1] 499527
```

```r
############################################
#####################################################
# Plot prices

asks <- as.numeric(tqdata$OFR)
bids <- as.numeric(tqdata$BID)
mids <- 0.5*bids + 0.5*asks

pmin = min(as.numeric(tqdata$PRICE))
pmax = max(as.numeric(tqdata$PRICE))
plot(as.numeric(tqdata$PRICE),col="red", type="l", ylab="Trade price",
     xlab="Trade #", main="Trade price (9:30-16:00)", ylim=c(pmin-0.1,pmax+0.1))
lines(mids, type="l", col="blue")
```




```
plot(as.numeric(tqdata$SIZE),col="red", type="l",
     ylab="Trade size",
     xlab="Trade #", main="Trade volume", ylim=c(0,100000))
```



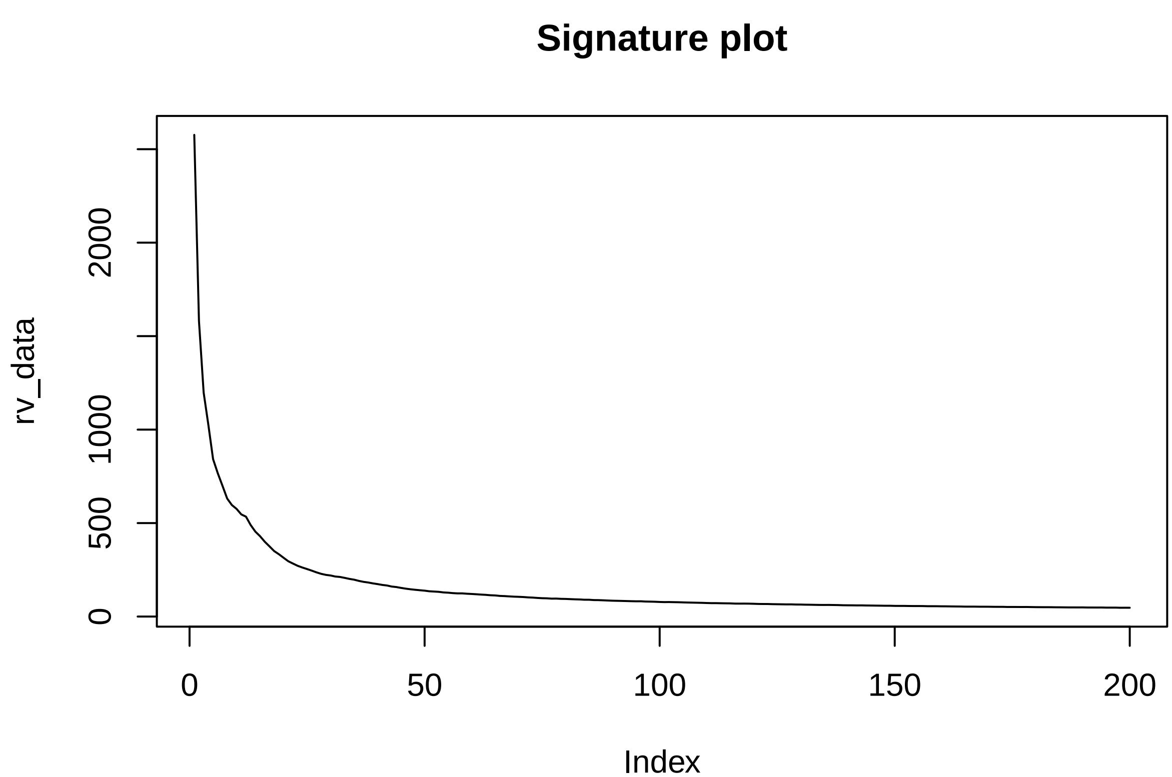
```r
#################################################################

# Draw the signature plot = RV(sampling freq)
# Signature plot = plot of RV as a function of the sampling freq

p <- as.numeric(tqdata$PRICE)

realizedVar <- function(q){rCov(diff(p, lag=q, differences=1))/q}

realizedVar(1)
```

```
## [1] 2576.673
```

```r
sqrt(realizedVar(1))
```

```
## [1] 50.76094
```

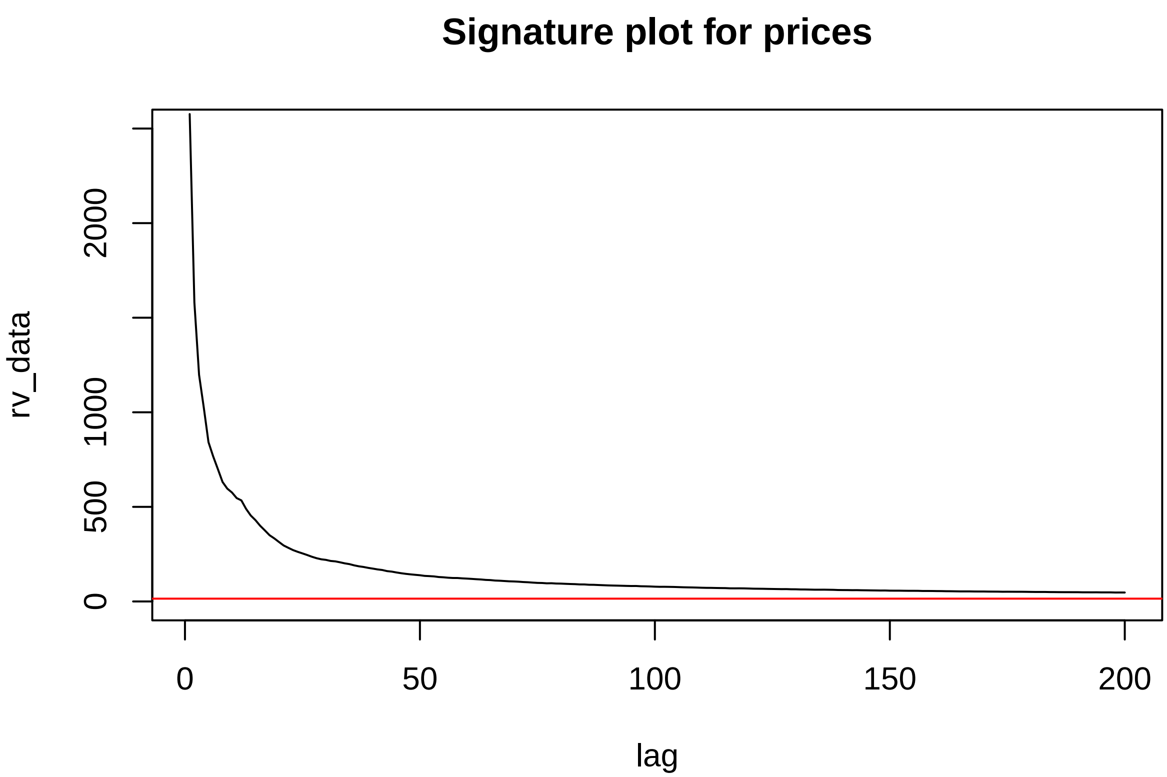
```r
# compute the signature plot RV(lag)

rv_data <- NULL

for(q in 1:200){

  rv_data <- c(rv_data, realizedVar(q))

}

plot(rv_data, type ="l", main="Signature plot")
```



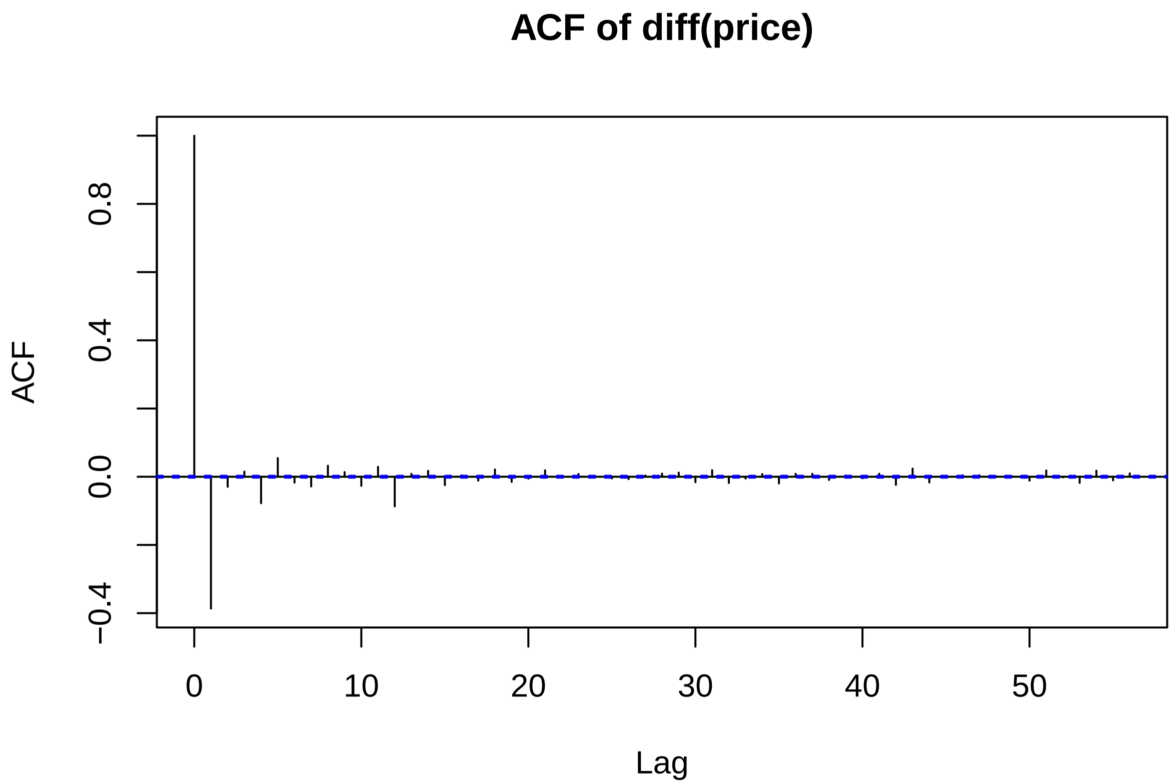
```r
# q5min is the number of trades per 5 mins.
# Compute q5min = n(trades)/5mins.
# Hint: there are 390 mins in a trading day
# Use it to compute the realized variance by sampling every 5 mins

n.trades <- dim(tqdata)[1]

q5min <- n.trades*5/390

  rv5 = realizedVar(q5min)

sqrt(rv5)
```

```
## [1] 3.841995
```

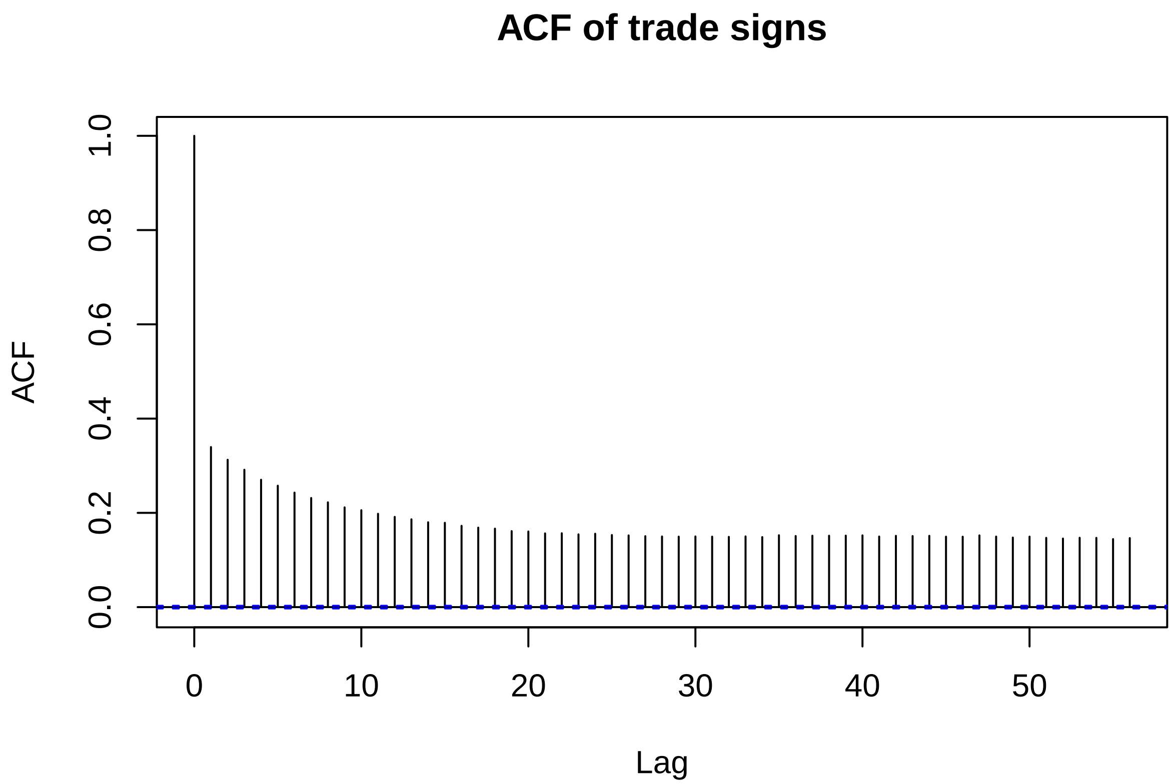
```r
#sqrt(252)*sqrt(rv5)/av.price


plot(rv_data, type ="l", main="Signature plot for prices",
     ylim=c(0,2500), xlab="lag")
abline(h=rv5,col="red")
```



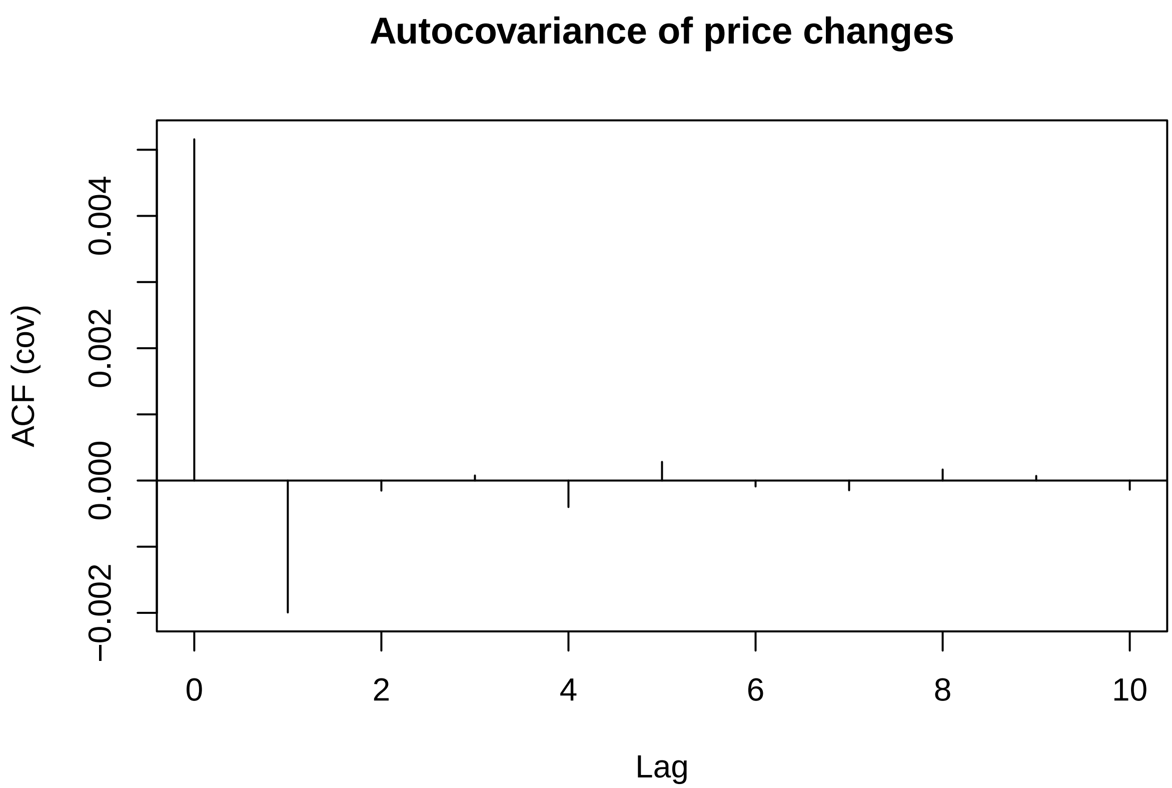

```r
####################################################################
# Now Roll estimate of volatility
# autocorrelation of price changes
dp <- diff(p)

acf(dp, main="ACF of diff(price)")
```



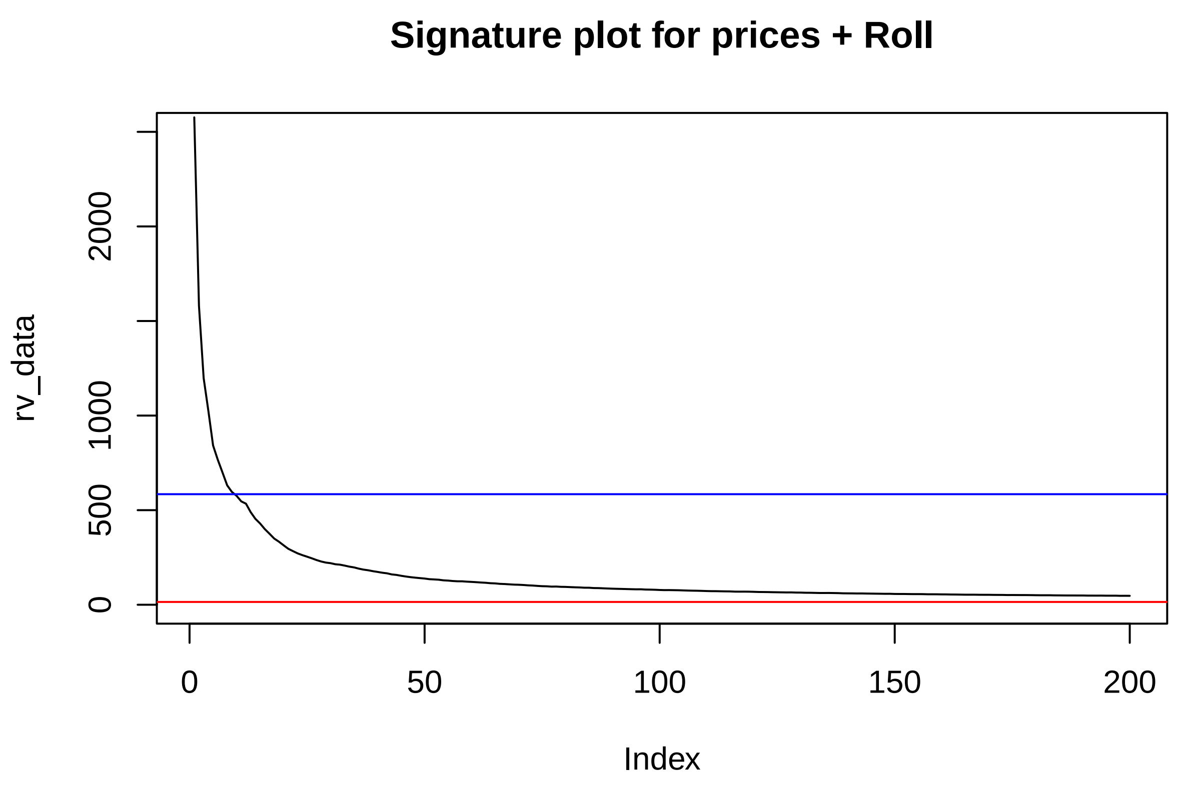

```r
# autocorrelation of trade signs
ts <- getTradeDirection(tqdata)
acf(ts, main="ACF of trade signs")
```



```r
###########################################################################
# Roll model estimate of the volatility the tqdata
# recall p are the trade prices, dp = p(t) - p(t-1) are price changes

dp = diff(p)

# compute the covariance of the price changes, for the Roll model analysis
covdp <- acf(dp, lag.max=10,
            type="covariance", plot=TRUE,
            main="Autocovariance of price changes")
```




```r
gamma0 <- covdp$acf[1]
gamma1 <- covdp$acf[2]

sig2u = gamma0 + 2*gamma1

rvRoll <- sig2u*n.trades

sigRoll <- sqrt(sig2u*n.trades)

plot(rv_data, type ="l",
     main="Signature plot for prices + Roll",
      ylim=c(0,2500))
abline(h=rv5,col="red")
abline(h=rvRoll,col="blue")
```




```
#############################################################
# Decompose the total volatility (gamma0) into
# i) Fundamental gammma0 + 2*gamma1
# ii) Trading noise (-2*gamma1)

n.trades
```

```
## [1] 499527
```

```
av.price <- mean(p)
av.price   # average trade price = 139.25
```

```
## [1] 218.3605
```

```
#daily volatility
sig.day <- sqrt(sig2u*n.trades)
sig.day    #2.679
```

```
## [1] 24.17668
```

```
#annualized volatility
sig.ann <- sqrt(252)*sig.day
sig.ann   # 42.54
```

```
## [1] 383.7929
```

```r
# log-normal volatility
sig.ann.ln <- sig.ann/av.price
sig.ann.ln     #30.55%
```

```
## [1] 1.757611
```

```r
# compare with the total volatility
sig.day.total <- sqrt(gamma0*n.trades)
sig.day.total    #7.22
```

```
## [1] 50.76099
```

```r
sig.ann.total <- sqrt(252)*sig.day.total
sig.ann.total
```

```
## [1] 805.8057
```

```r
sig.ann.ln.total <- sig.ann.total/av.price
sig.ann.ln.total  # 82.35%
```

```
## [1] 3.690253
```

3. Estimate the probability of informed trading (PIN measure)

```r
# count B/S events

x <- getTradeDirection(tsla.taq.adf)

tradeDirection <- matrix(x)

buy_side <- which(tradeDirection >0)

num_buy_side <- length(matrix(buy_side))
num_sell_side <- length(tradeDirection) - length(matrix(buy_side))

ntrades <- cbind(num_buy_side, num_sell_side)

ntrades
```

```
##      num_buy_side num_sell_side
## [1,]       343263        156264
```

```r
# run optimization of likelihood function

Buy <- c(350,250,500,552)
Sell <- c(382, 500, 463, 550)
data = cbind(Buy,Sell)
```

```r
par0 = c(0.5,0.5,300,400,500)


# Call EHO function
EHO_out = EHO(data)
model = optim(par0, EHO_out, gr = NULL,
              method = c("BFGS"), hessian = FALSE)

model
```

```
## $par
## [1]    0.4742337 -167.5626061  225.0306286  300.4342112  473.6772463
##
## $value
## [1] -18148.33
##
## $counts
## function gradient
##      122      100
##
## $convergence
## [1] 1
##
## $message
## NULL
```

```r
## Parameter Estimates
model$par[1] # Estimate for alpha
```

```
## [1] 0.4742337
```

```r
model$par[2] # Estimate for delta
```

```
## [1] -167.5626
```

```r
model$par[3] # Estimate for mu
```

```
## [1] 225.0306
```

```r
model$par[4] # Estimate for eb
```

```
## [1] 300.4342
```

```r
model$par[5] # Estimate for es
```

```
## [1] 473.6772
```

```r
## Estimate for PIN
(model$par[1]*model$par[3])/((model$par[1]*model$par[3])+model$par[4]+model$par[5])
```

```
## [1] 0.1211554
```