# 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**

```
trades<-read.csv("trades_ETHUSD.csv")
quotes<-read.csv("quotes_ETHUSD.csv")


quotes$timestamp<-ymd_hms(quotes$timestamp)
quotes <- as.data.table(quotes)

trades$timestamp<-ymd_hms(trades$timestamp)
trades <- as.data.table(trades)


trades<-trades%>%mutate(DT=timestamp)%>%select(DT,symbol,size,price,tickDirection)
quotes<-quotes%>%mutate(BID=bidPrice,ASK=askPrice,DT=timestamp)%>%select(DT,symbol,bidSize,BID,askSize,
taq.ethusd<-matchTradesQuotes(trades,quotes)
```

```
# Calculate effective spread
quoted_spread <- taq.ethusd$OFR - taq.ethusd$BID

# Calculate midpoint
midpoint <- (taq.ethusd$OFR + taq.ethusd$BID) / 2

# Calculate effective spread
effective_spread <- 2 * abs(taq.ethusd$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 * (taq.ethusd$PRICE - future_midpoint)

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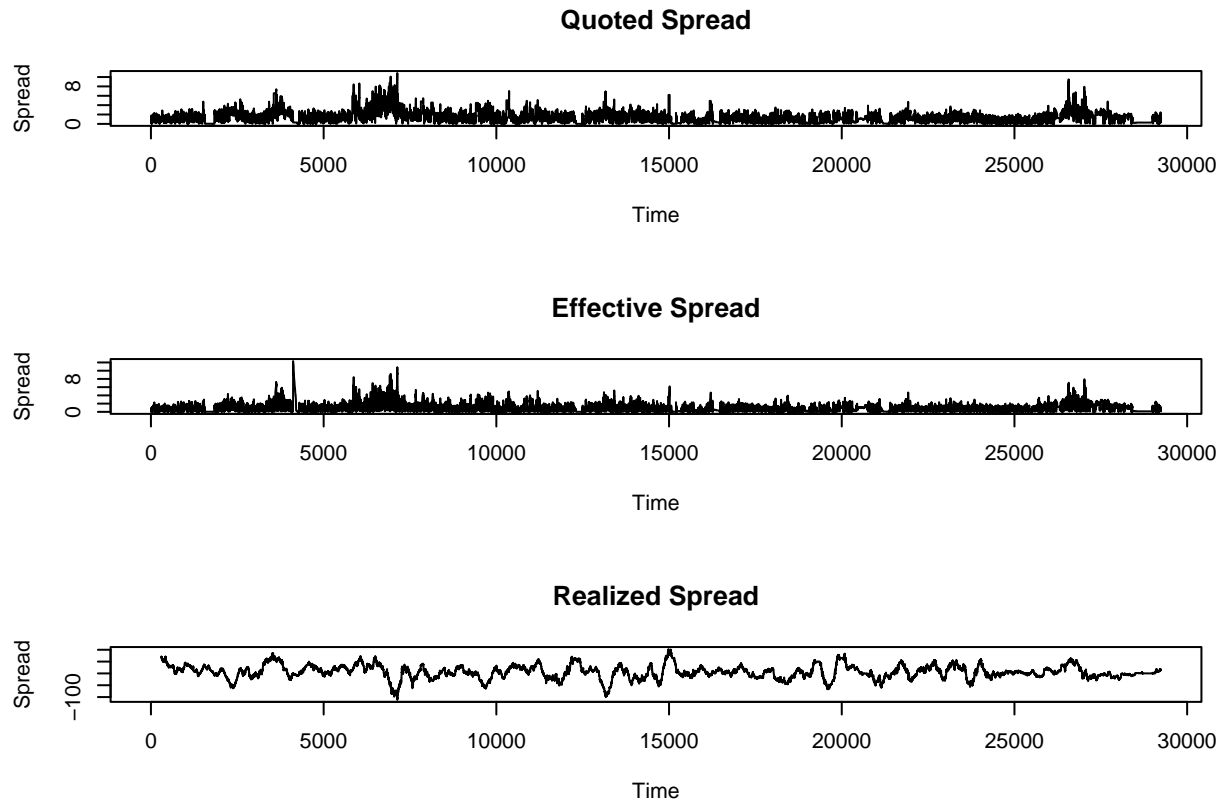
```r
par(mfrow = c(3, 1))  # Split plots
plot(quoted_spread, type = "l", main = "Quoted Spread", ylab = "Spread", xlab = "Time")
plot(effective_spread, type = "l", main = "Effective Spread", ylab = "Spread", xlab = "Time")
plot(realized_spread, type = "l", main = "Realized Spread", ylab = "Spread", xlab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

**Quoted Spread**



**Effective Spread**



**Realized Spread**



## 2. Estimate the volatility using intraday data

```r
# loads a xts file called tqdataMktHrs
tqdata <- taq.ethusd

head(tqdata)
```

```
##                      DT SYMBOL BIDSIZ      BID OFRSIZ      OFR SIZE    PRICE
## 1: 2024-11-06 00:00:05 ETHUSD     26 2423.32     53 2423.94    3 2423.63
## 2: 2024-11-06 00:00:17 ETHUSD     29 2422.51     17 2423.32    9 2423.32
## 3: 2024-11-06 00:00:17 ETHUSD     29 2422.51      9 2423.32    8 2423.32
## 4: 2024-11-06 00:00:27 ETHUSD     14 2422.69     28 2423.50   12 2422.69
## 5: 2024-11-06 00:00:27 ETHUSD      8 2422.94     26 2423.75    7 2422.94
## 6: 2024-11-06 00:00:28 ETHUSD     24 2422.96     53 2423.81    3 2422.97
##     TICKDIRECTION
## 1:      PlusTick
## 2:     MinusTick
## 3: ZeroMinusTick
## 4:     MinusTick
## 5:      PlusTick
## 6:      PlusTick
```

```r
tail(tqdata)
```

```
##                        DT SYMBOL BIDSIZ     BID OFRSIZ     OFR SIZE   PRICE
## 1: 2024-11-06 23:59:37 ETHUSD     32 2727.24     52 2727.59    5 2727.30
## 2: 2024-11-06 23:59:38 ETHUSD     32 2727.24     52 2727.96    1 2727.43
## 3: 2024-11-06 23:59:39 ETHUSD     27 2727.61     69 2728.52    1 2727.67
## 4: 2024-11-06 23:59:57 ETHUSD     32 2726.88    217 2728.54    1 2727.63
## 5: 2024-11-06 23:59:58 ETHUSD     32 2726.88     56 2728.53    1 2727.83
## 6: 2024-11-06 23:59:58 ETHUSD     15 2727.72      1 2728.03    1 2727.72
##     TICKDIRECTION
## 1:  ZeroPlusTick
## 2:      PlusTick
## 3:      PlusTick
## 4:     MinusTick
## 5:      PlusTick
## 6:     MinusTick
```

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

```
## [1] 29244
```

```r
# summarize trades by exchange

length(tqdata$SIZE)
```
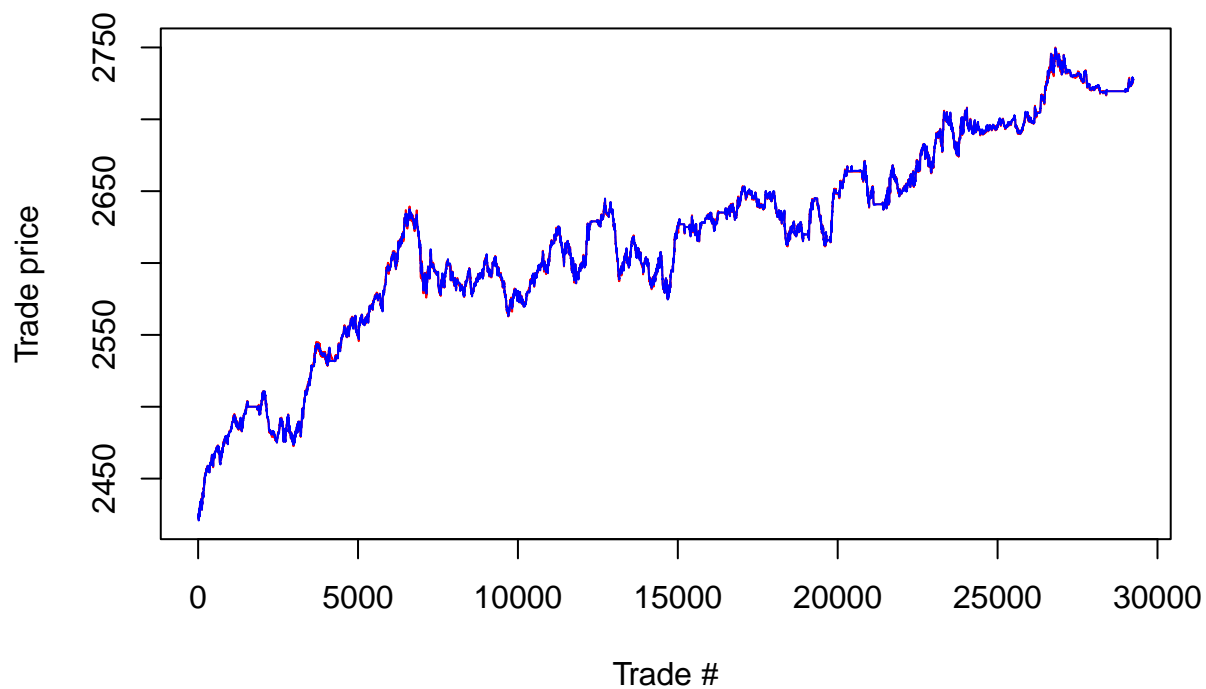
```
## [1] 29244
```

```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")
```
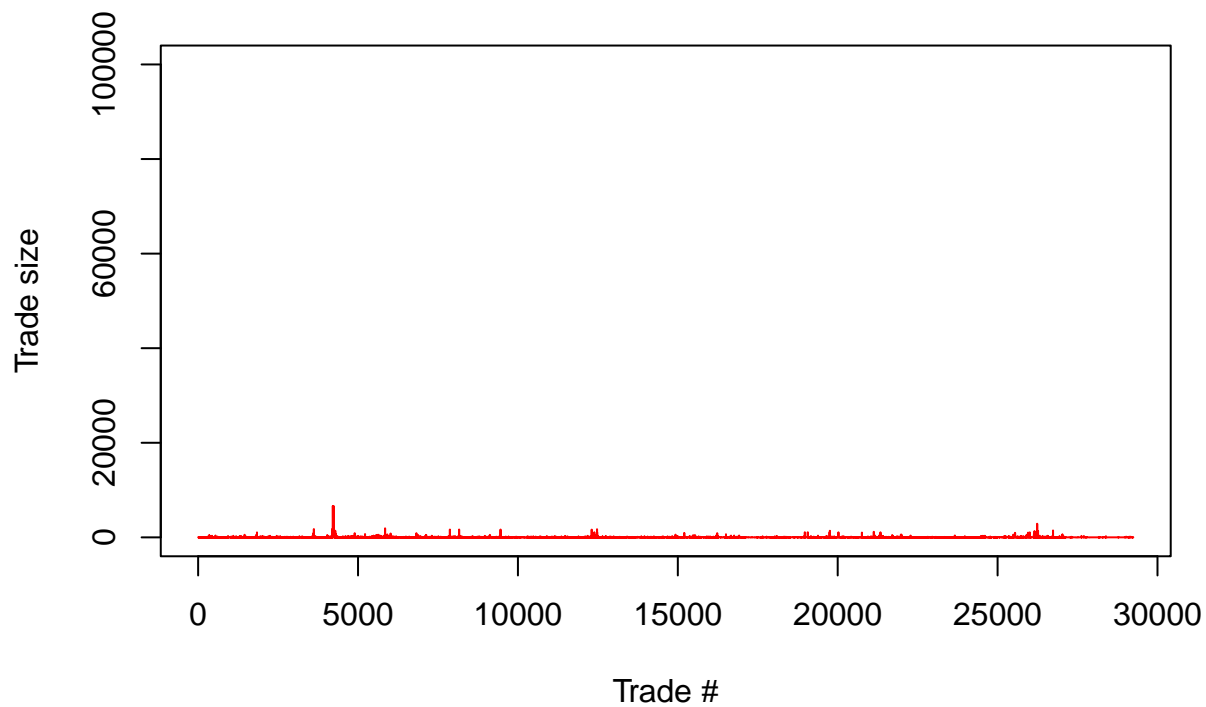
## Trade price (9:30–16:00)



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

## Trade volume

```
##############################################################

# 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] 14895
```

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

```
## [1] 122.0451
```

```
# 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")
```
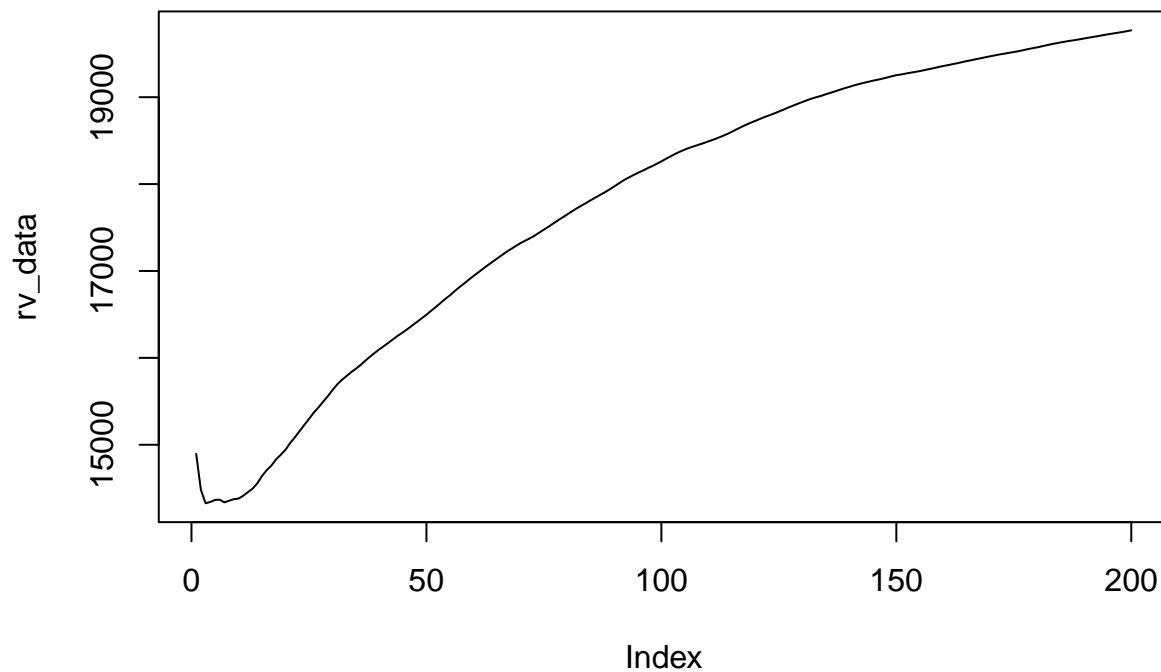
**Signature plot**

```
# 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] 139.933
```

```
#sqrt(252)*sqrt(rv5)/av.price
```
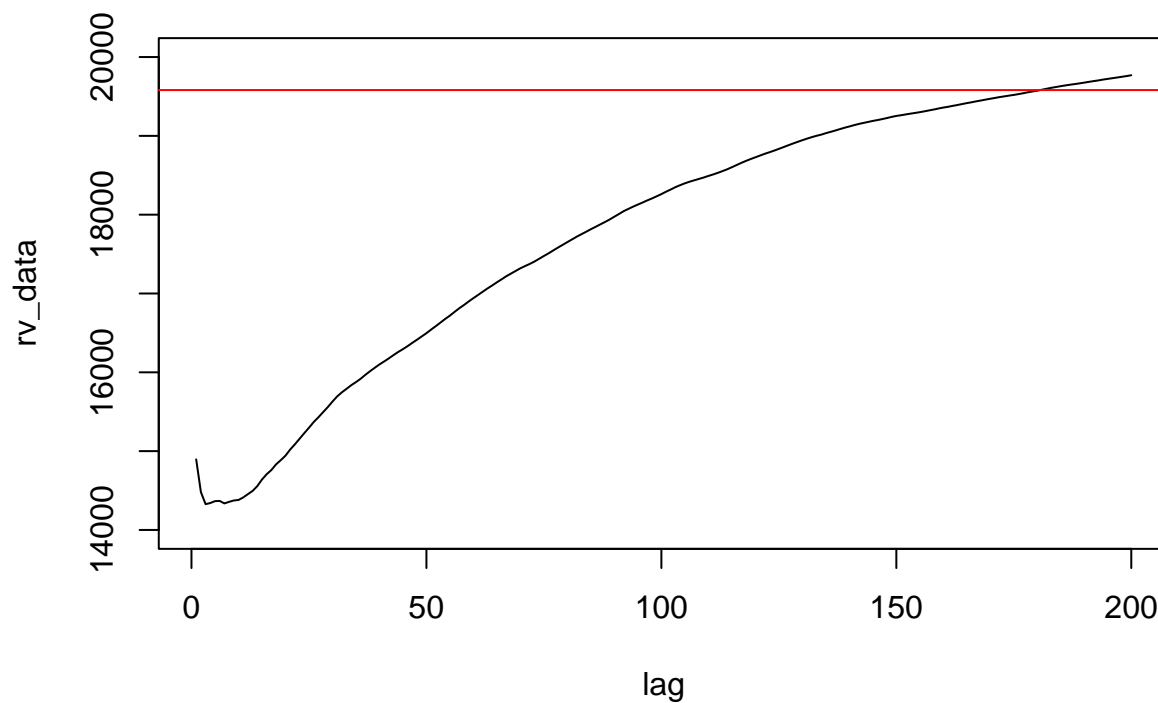
```
plot(rv_data, type ="l", main="Signature plot for prices",
     ylim=c(14000,20000), xlab="lag")
abline(h=rv5,col="red")
```
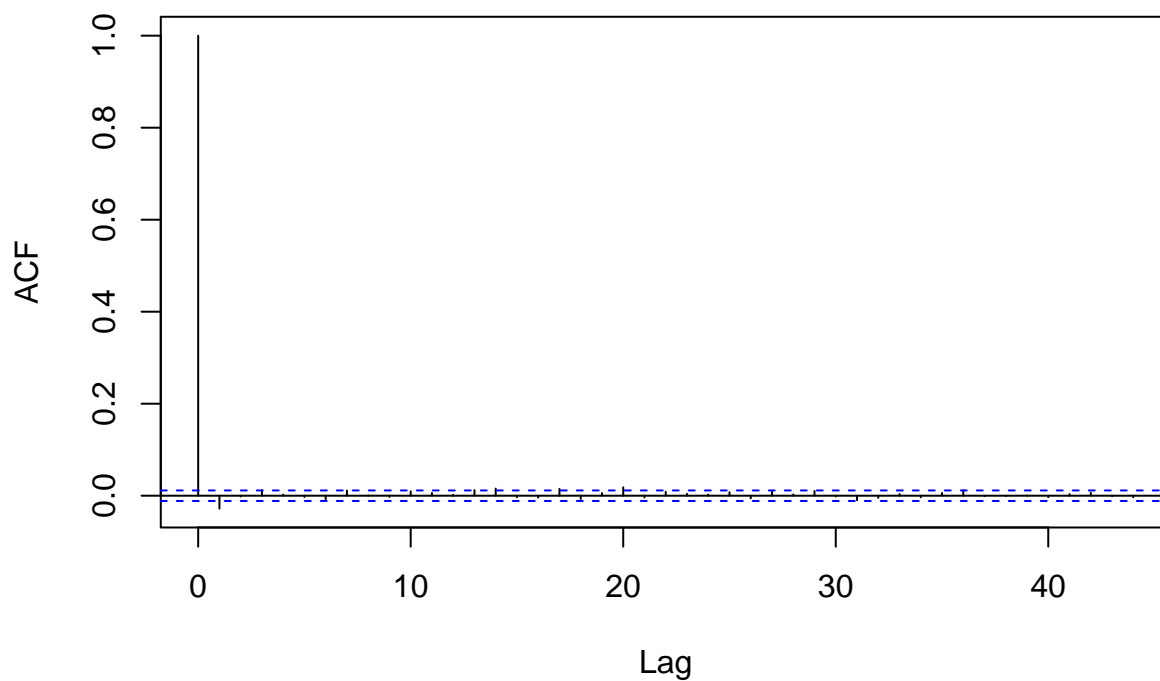
## Signature plot for prices



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

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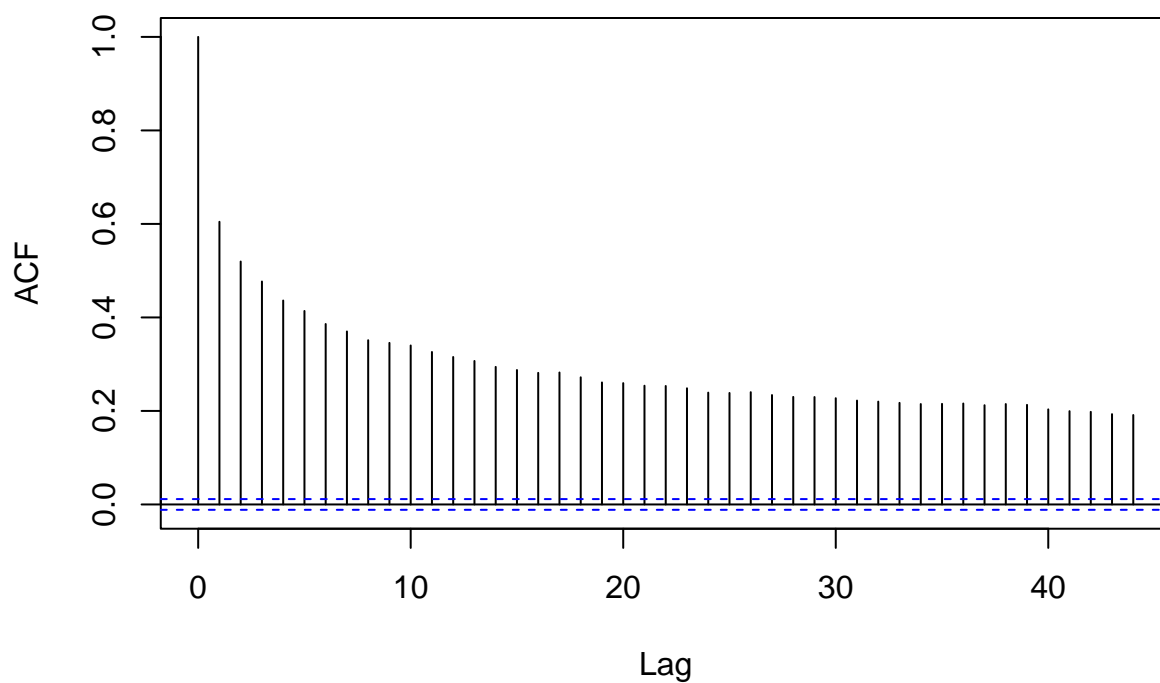
## ACF of diff(price)



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

## ACF of trade signs

```
###################################################################
# 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")
```
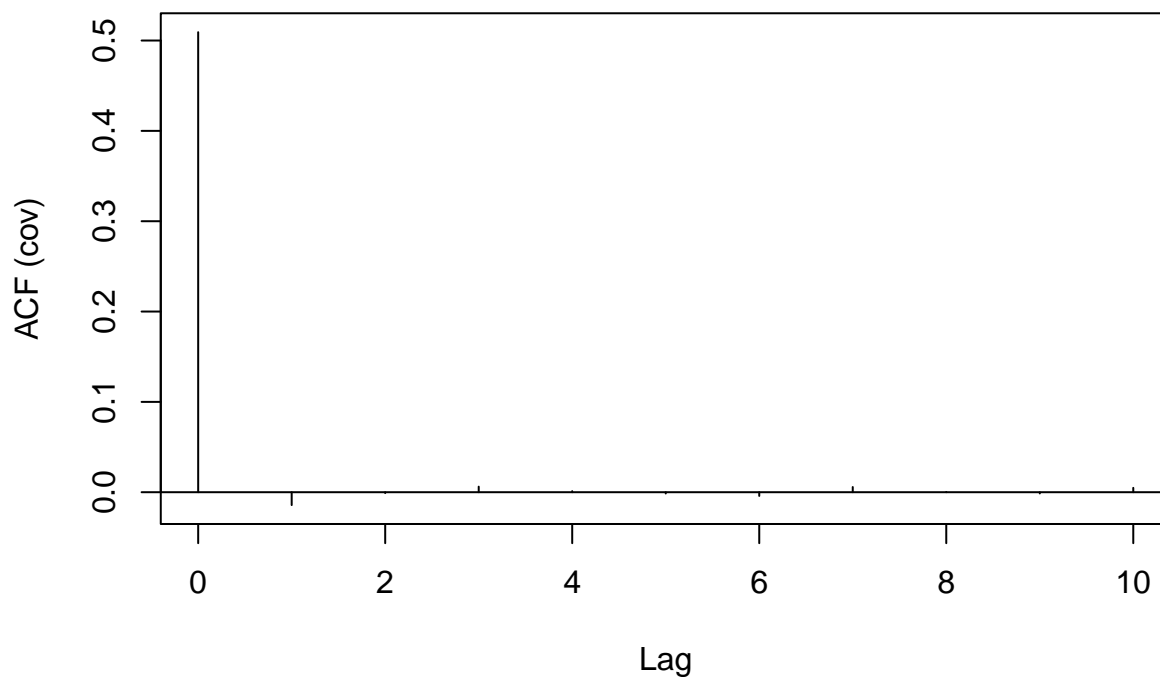
## Autocovariance of price changes



```
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(14000,20000))
abline(h=rv5,col="red")
abline(h=rvRoll,col="blue")
```
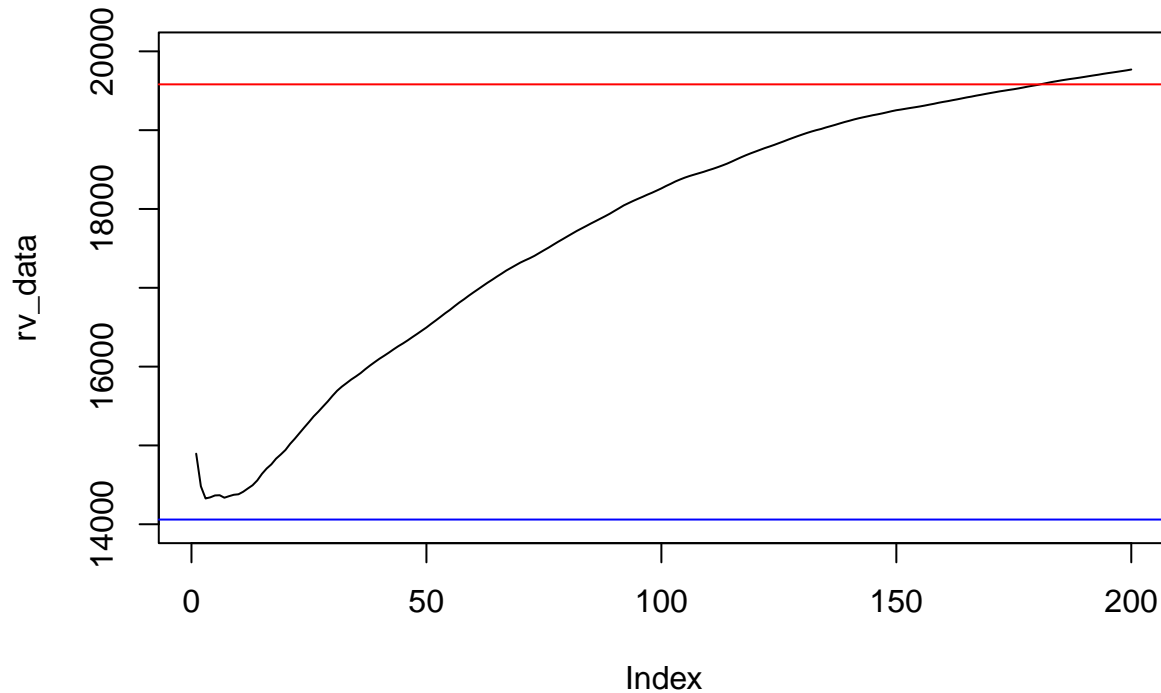
# Signature plot for prices + Roll



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

n.trades
```

```
## [1] 29244
```

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

```
## [1] 2617.066
```

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

```
## [1] 118.5727
```

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

```
## [1] 1882.283
```

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

```
## [1] 0.7192341
```

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

```
## [1] 122.0342
```

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

```
## [1] 1937.233
```

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

```
## [1] 0.7402308
```

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

```
# count B/S events

x <- getTradeDirection(taq.ethusd)

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,]        13314         15930
```

```
# run optimization of likelihood function

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

```
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
```

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

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

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

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

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

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

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

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

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

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

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

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