quantstrat II: Order Sizing

CFRM 522 (Copyright 2021 Daniel Hanson)

May 5, 2021

Before looking at a more complex example involving Bollinger Bands and short sales, let's look at a simpler example to introduce order sizing in quantstrat

For this example, we will use a contrived strategy as follows:

– Buy 100 shares whenever the daily high price exceeds the 5-day moving average

– Completely exit when the daily closing price drops below the 5-day moving average

– Note that this allows pyramiding on the buy side, in that we can build up exposure in the equity or ETF.

We will first look at the unconstrained case

We will then examine how to impose a 200 share limit using quantstrat commands

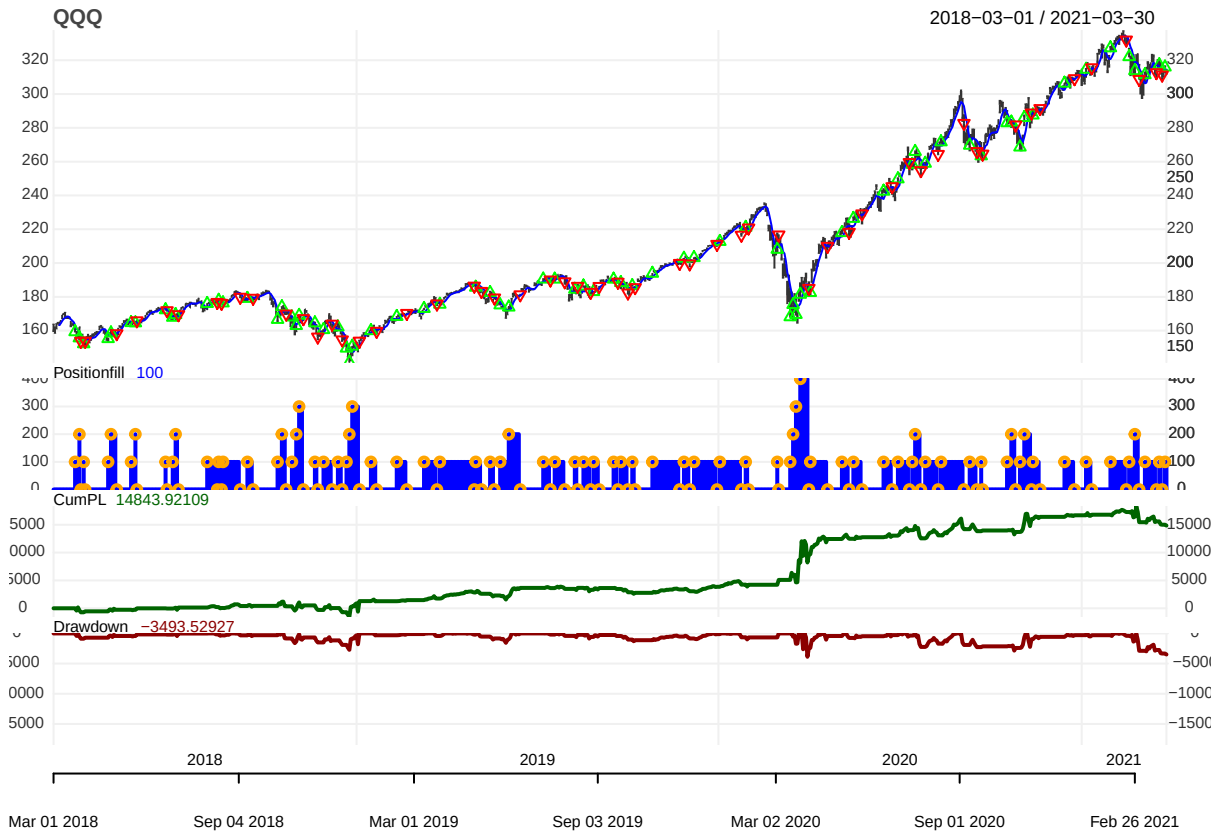
```r
library(quantstrat)

# See order_sizing_015.R

symbol <- 'QQQ'

currency("USD")
stock(symbol, currency="USD", multiplier=1)

# Fetch historical data:
initDate <- '2018-02-28'
startDate <- '2018-03-01'
endDate <- '2021-03-31'
initEq <- 1000000   # $1M
numShares <- 100
n <- 5    # Number of days in moving average

Sys.setenv(TZ="UTC")
getSymbols(symbol, from=startDate, to=endDate, index.class="POSIXct", adjust=TRUE)
etfData <- get(symbol)

portName <- "smaCross"
stratName <- portName
acctName <- portName

suppressWarnings(rm.strat(stratName))
```

```
initPortf(name = portName, symbols = symbol, initDate = initDate,
          currency = 'USD')

initAcct(name = acctName, portfolios = portName,
         initDate=initDate, initEq=initEq)

initOrders(portfolio = portName, initDate = initDate)
strategy(name = stratName, store=TRUE)

# Add an indicator; SMA(.) is a TTR function
add.indicator(strategy = stratName, name = "SMA",
              arguments = list(x = quote(Cl(mktdata)), n=n), label="SMA_")

# There are two signals:
# The first is when daily high price crosses over the n-day SMA
add.signal(strategy = stratName, name = "sigCrossover",
           arguments = list(columns=c("High","SMA_"),relationship="gte"),
           label="Hi.gt.SMA")

# The second is when the daily closing price crosses under the n-day SMA
add.signal(strategy = stratName, name="sigCrossover",
           arguments = list(columns=c("Close","SMA_"),relationship="lt"),
           label="Cl.lt.SMA")
```

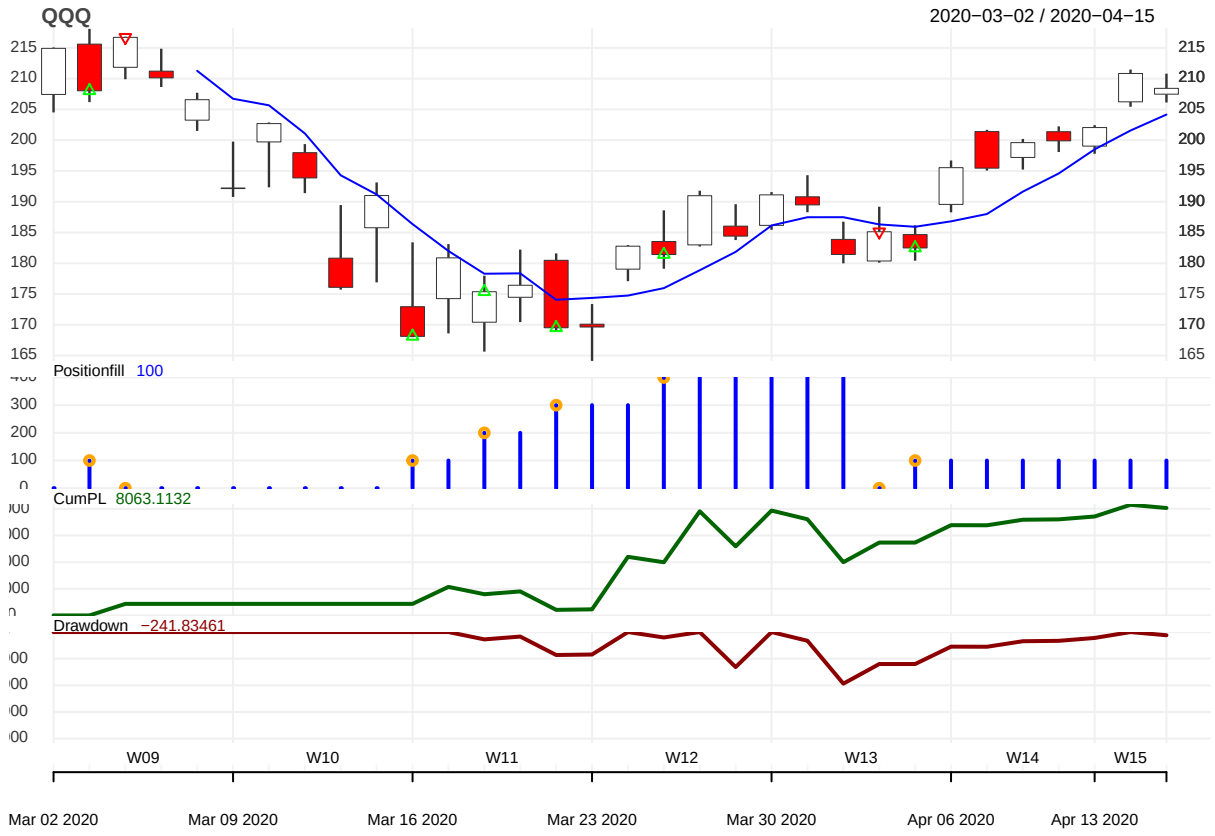
```
# There are two rules:
# The first is to buy when the high price crosses above the SMA
add.rule(strategy = stratName, name='ruleSignal',
         arguments = list(sigcol="Hi.gt.SMA", sigval=TRUE, orderqty=numShares,
                          ordertype='market', orderside='long', pricemethod='market',
                          TxnFees = 0), type = 'enter', path.dep = TRUE)

# The second is to sell when the closing price crosses below the SMA
add.rule(strategy = stratName, name='ruleSignal',
         arguments = list(sigcol="Cl.lt.SMA", sigval=TRUE, orderqty='all',
                          ordertype='market', orderside='long', pricemethod='market',
                          TxnFees = 0), type = 'exit', path.dep = TRUE)

# Process the indicators and generate trades
applyStrategy(strategy = stratName, portfolios = portName)

updatePortf(Portfolio = portName)
updateAcct(name = acctName)
updateEndEq(Account = acctName)
```

Now, we can look at a couple of the `chart.Posn(.)` plots, and then use them as references to compare with the order sizing cases that will follow.

```
.TA = paste("add_SMA(n =", n, ", col = 'blue')")
chart.Posn(Portfolio = portName, TA = .TA)
```



```
chart.Posn(Portfolio = portName, TA = .TA,
           Dates = '2020-03-01::2020-04-15')
```



We see there are three cases where the equity exposure went above 200 shares.

The case during March 2020 can be seen in more detail in the 2nd plot above.

In the next example, we will modify the previous code and impose a 200 share holding limit.

First, however, we need to examine the mechanism for doing so.

The process starts inside the `add.rule(.)` rule definition for entering the market. We need to add the `osFUN` argument for the embedded `ruleSignal(.)` function, and set it to `osMaxPos`, as shown here:

```
add.rule(strategy = stratName, name='ruleSignal',
         arguments = list(sigcol="Hi.gt.SMA", sigval=TRUE, orderqty=numShares,
                          ordertype='market', orderside='long', pricemethod='market',
                          TxnFees = 0, osFUN=osMaxPos), type = 'enter', path.dep = TRUE)
```

`osMaxPos`, however, depends on the existence of a call to the function `addPosLimit(.)`, with the appropriate arguments, following the `add.rule(.)` statement; viz,

```
# This limits our exposure to 200 shares:
addPosLimit(portName, symbol, timestamp=initDate, maxpos=200, minpos=0)
```

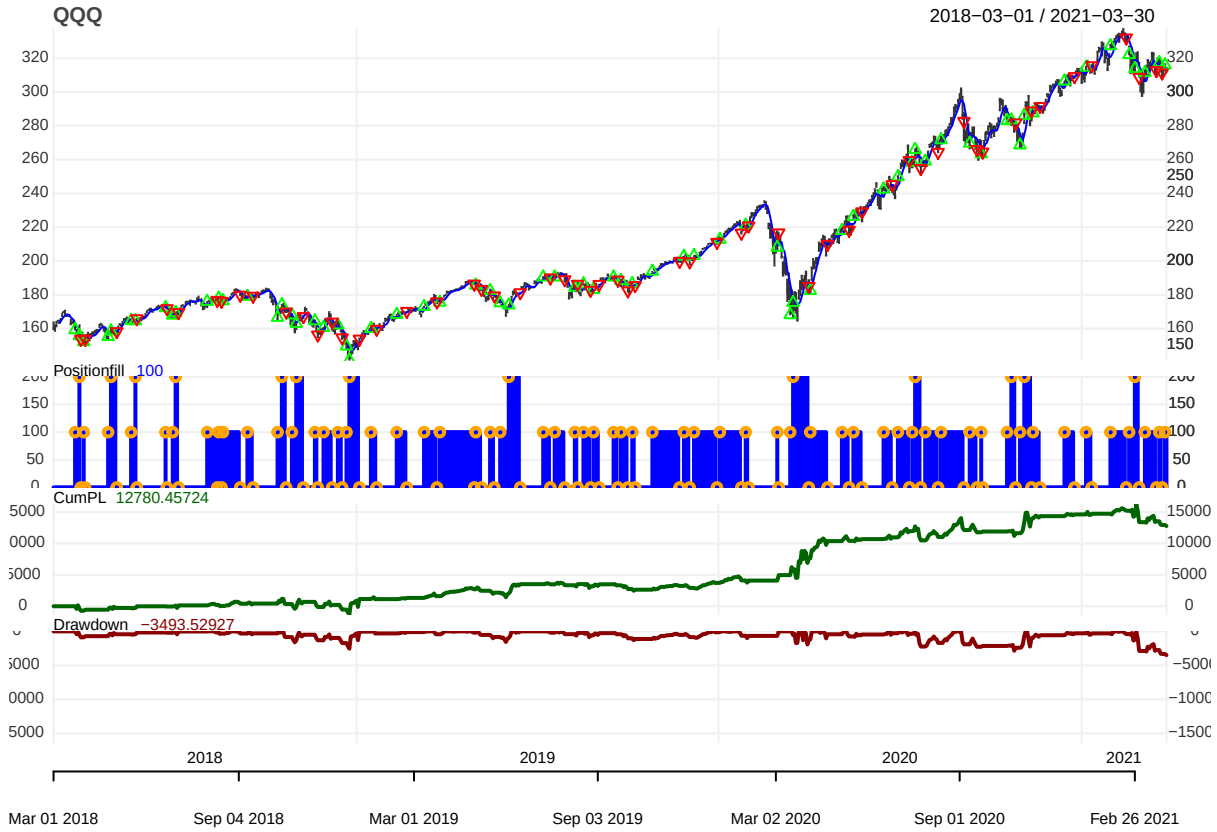
Now, let's make this modification and run the code again.

```
portName <- "smaCrossSizeLmt"
stratName <- portName
acctName <- portName

suppressWarnings(rm.strat(stratName))

initPortf(name = portName, symbols = symbol, initDate = initDate,
          currency = 'USD')

initAcct(name = acctName, portfolios = portName,
         initDate=initDate, initEq=initEq)

initOrders(portfolio = portName, initDate = initDate)
strategy(name = stratName, store=TRUE)

# Add an indicator; SMA(.) is a TTR function
add.indicator(strategy = stratName, name = "SMA",
              arguments = list(x = quote(Cl(mktdata)), n=n), label="SMA_")

# There are two signals:
# The first is when daily high price crosses over the n-day SMA
add.signal(strategy = stratName, name = "sigCrossover",
           arguments = list(columns=c("High","SMA_"),relationship="gte"),
           label="Hi.gt.SMA")

# The second is when the daily closing price crosses under the n-day SMA
add.signal(strategy = stratName, name="sigCrossover",
           arguments = list(columns=c("Close","SMA_"),relationship="lt"),
           label="Cl.lt.SMA")
```

```r
# There are two rules:
# The first is to buy when the high price crosses above the SMA
add.rule(strategy = stratName, name='ruleSignal',
         arguments = list(sigcol="Hi.gt.SMA", sigval=TRUE, orderqty=numShares,
                          ordertype='market', orderside='long', pricemethod='market',
                          TxnFees = 0, osFUN=osMaxPos), type = 'enter', path.dep = TRUE)

# The second is to sell when the closing price crosses below the SMA
add.rule(strategy = stratName, name='ruleSignal',
         arguments = list(sigcol="Cl.lt.SMA", sigval=TRUE, orderqty='all',
                          ordertype='market', orderside='long', pricemethod='market',
                          TxnFees = 0), type = 'exit', path.dep = TRUE)

# This limits our exposure to 200 shares:
addPosLimit(portName, symbol, timestamp=initDate, maxpos=200, minpos=0)

# Process the indicators and generate trades
applyStrategy(strategy = stratName, portfolios = portName)

updatePortf(Portfolio = portName)
updateAcct(name = acctName)
updateEndEq(Account = acctName)
```

Now, let's look at the same `chart.Posn(.)` plots, and compare with the earlier case with no order sizing.

```
.TA = paste("add_SMA(n =", n, ", col = 'blue')")
chart.Posn(Portfolio = portName, TA = .TA)
```



```
chart.Posn(Portfolio = portName, TA = .TA,
           Dates = '2020-03-01::2020-04-15')
```

Note now that we never exceed a holding of 200 shares anywhere in the plots.

We will see a more sophisticated example involving Bollinger Bands and short sales in CFRM522_015(G)_quantstratII.pdf, but the concept is the same.