quantstrat introduction

CFRM 522 (Copyright 2021 Daniel Hanson, Guy Yollin)

04 May, 2021

Continuing with the Faber single-fund XLU example

We previously set up and ran the pre- and post- backtest functions using the blotter R package. We will start with the same premise, but now concentrate on setting up and running backtests using the quantstrat package

```
library(quantstrat)
symbol <- "XLU"
currency("USD")
stock(symbol, currency="USD", multiplier=1)

# Fetch historical data:
initDate <- '1997-12-31'
startDate <- '1998-01-01'
endDate <- '2018-03-31'
initEq <- 1000000   # $1M
numShares <- 19000  # rounded lot closest to $1M

Sys.setenv(TZ="UTC")
getSymbols.yahoo(symbol, from=startDate, to=endDate,
           index.class="POSIXct", adjust=TRUE,
```

```
            periodicity = "monthly", env = ".GlobalEnv")

etfData <- get(symbol)  # Need this for the dual nature
                        # of the quantmod symbol object.
```

Next, continue with the blotter setup: portfolio and account:

```
portfName <- "faberXlu"
initPortf(name = portfName, symbols = symbol, initDate = initDate,
          currency = 'USD')

acctName <- portfName
initAcct(name = acctName, portfolios = portfName,
         initDate=initDate, initEq=initEq)
```

The necessary preliminary steps involving blotter commands is complete. We can now move to quantstrat. First, some initialization and setup steps. The initialization consists of

1. **initializing order container with `initOrders(.)`**
2. **calling the `strategy(.)` constructor to create a strategy object**

```
# Initialize orders and strategy objects
stratName <- portfName    # Just use same name for strategy
initOrders(portfolio = portfName, initDate = initDate)
strategy(name = stratName, store=TRUE)
```

We now define the quantstrat components that will be necessary to run the backtest, namely

call `add.indicator(.)` to add an indicator to the strategy

call `add.signal(.)` add the signals to the strategy (there are two in this case)

```
# Add an indicator; SMA(.) is a TTR function
add.indicator(strategy = stratName, name = "SMA",
              arguments = list(x = quote(Cl(mktdata)), n=10), label="SMA10")


# There are two signals:
# The first is when monthly price crosses over the 10-month SMA
add.signal(strategy = stratName, name = "sigCrossover",
           arguments = list(columns=c("Close","SMA10"),relationship="gte"),
           label="Cl.gt.SMA")


# The second is when the monthly price crosses under the 10-month SMA
add.signal(strategy = stratName, name="sigCrossover",
           arguments = list(columns=c("Close","SMA10"),relationship="lt"),
           label="Cl.lt.SMA")
```

call `add.rule(.)` add the rules to the strategy (there are again two in this case)

```
# There are two rules:
# The first is to buy when the price crosses above the SMA
add.rule(strategy = stratName, name='ruleSignal',
         arguments = list(sigcol="Cl.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 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)
```

The strategy object contains:

1 user defined indicator

2 user defined signals

2 user defined trading rules

We now run the backtest by calling the quantstrat function `applyStrategy(.)`

Notice that the resulting trades in the output occur only at end-of-month.

```
applyStrategy(strategy = stratName, portfolios = portfName)
```

```
## [1] "2000-04-01 00:00:00 XLU 19000 @ 28.484375"
## [1] "2000-06-01 00:00:00 XLU -19000 @ 27.25"
## [1] "2000-08-01 00:00:00 XLU 19000 @ 28.75"
## [1] "2001-06-01 00:00:00 XLU -19000 @ 31.15"
## [1] "2003-04-01 00:00:00 XLU 19000 @ 19.99"
## [1] "2006-03-01 00:00:00 XLU -19000 @ 30.83"
## [1] "2006-06-01 00:00:00 XLU 19000 @ 32.290001"
## [1] "2007-07-01 00:00:00 XLU -19000 @ 38"
## [1] "2007-09-01 00:00:00 XLU 19000 @ 39.799999"
## [1] "2008-01-01 00:00:00 XLU -19000 @ 39.220001"
## [1] "2008-04-01 00:00:00 XLU 19000 @ 39.880001"
## [1] "2008-07-01 00:00:00 XLU -19000 @ 38.080002"
## [1] "2009-07-01 00:00:00 XLU 19000 @ 28.940001"
## [1] "2010-05-01 00:00:00 XLU -19000 @ 28.76"
## [1] "2010-07-01 00:00:00 XLU 19000 @ 30.370001"
## [1] "2012-11-01 00:00:00 XLU -19000 @ 35.32"
## [1] "2013-01-01 00:00:00 XLU 19000 @ 36.580002"
## [1] "2013-08-01 00:00:00 XLU -19000 @ 37.299999"
## [1] "2013-10-01 00:00:00 XLU 19000 @ 38.779999"
## [1] "2013-11-01 00:00:00 XLU -19000 @ 38.029999"
## [1] "2014-01-01 00:00:00 XLU 19000 @ 39.099998"
## [1] "2015-03-01 00:00:00 XLU -19000 @ 44.43"
## [1] "2016-01-01 00:00:00 XLU 19000 @ 45.419998"
```

```
## [1] "2016-11-01 00:00:00 XLU -19000 @ 46.75"
## [1] "2017-02-01 00:00:00 XLU 19000 @ 51.77"
## [1] "2017-12-01 00:00:00 XLU -19000 @ 52.68"
```

At this stage, we are done with running the strategy backtest in quantstrat

We will now return to blotter functions to process and examine the results

First, update the portfolio: this 'marks the book' of the portfolio with the various accounting measures. It goes through each symbol and calculates the PL for each period prices are available.

Following this, update the account: this performs the equity account calculations from the portfolio data and corresponding close prices. Remember: this requires that `updatePortf(.)` has already been run.

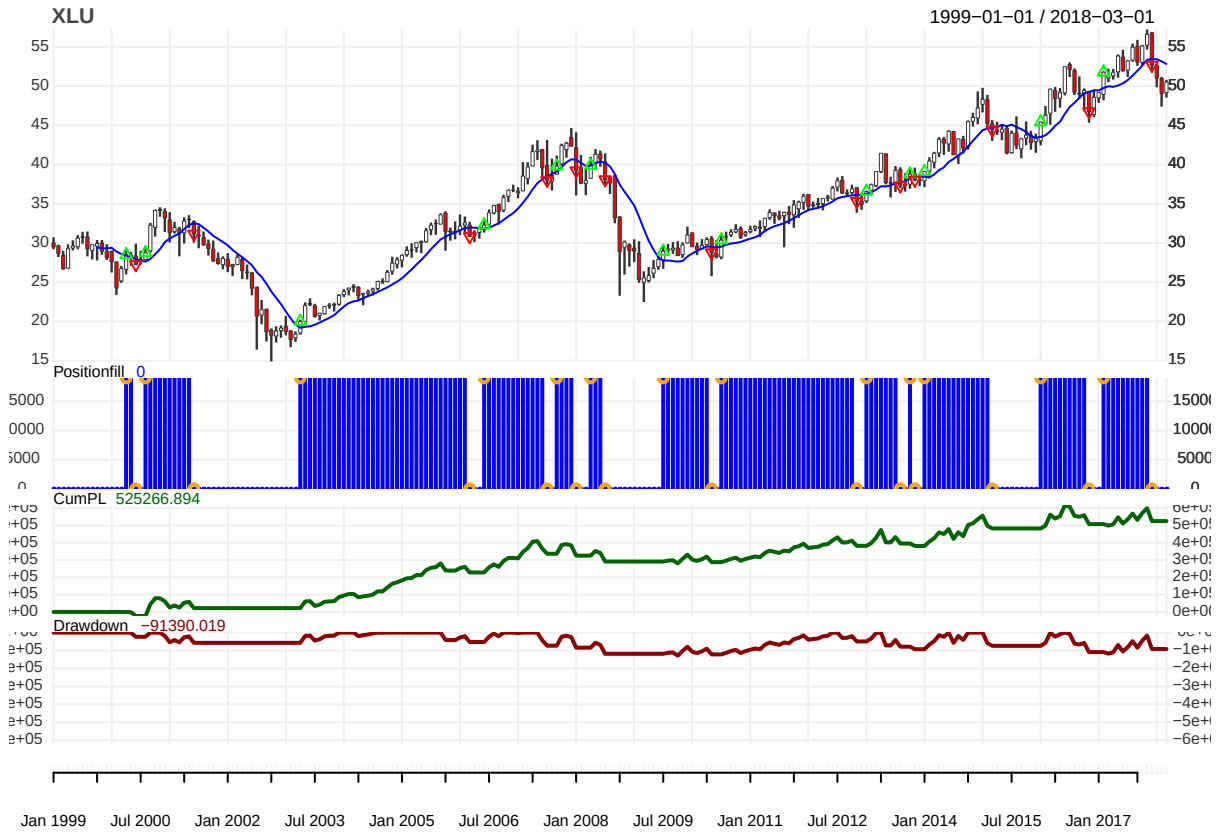
Finally, update ending equity for an account. Remember also: this requires that `updateAcct(.)` has already been run.

Note: For the analysis in this first example, we are only concerned with the portfolio level; however, in future examples and assignments, managing the account and ending equity will become important.

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

Results: First, here's a chart of trades against market data, position through time, and cumulative P & L

```
chart.Posn(Portfolio = portfName, Symbol = symbol,
           TA = "add_SMA(n = 10, col = 'blue')")
```



Next, view the record of transactions (`blotter::getTxns(.)`)

```
txns <- getTxns(Portfolio=portfName, Symbol=symbol)
txns['2007::2009']
```

```
##            Txn.Qty Txn.Price Txn.Fees Txn.Value Txn.Avg.Cost
## 2007-07-01  -19000     38.00        0   -722000        38.00
## 2007-09-01   19000     39.80        0    756200        39.80
## 2008-01-01  -19000     39.22        0   -745180        39.22
## 2008-04-01   19000     39.88        0    757720        39.88
## 2008-07-01  -19000     38.08        0   -723520        38.08
## 2009-07-01   19000     28.94        0    549860        28.94
##            Net.Txn.Realized.PL
## 2007-07-01           108489.98
## 2007-09-01                0.00
## 2008-01-01           -11019.96
## 2008-04-01                0.00
## 2008-07-01           -34199.98
## 2009-07-01                0.00
```

The mktdata object

`mktdata` is a special variable constructed during the execution of `applyStrategy(.)`. It is a time series object which contains the historic price data as well as the calculated indicators, signals, and rules

Inspecting `mktdata` can be very helpful in understanding strategy processing and debugging

```
mktdata['2008']
```

```
##            XLU.Open XLU.High XLU.Low XLU.Close XLU.Volume XLU.Adjusted
## 2008-01-01    42.15    44.11   36.05     39.22  184879000     24.08901
## 2008-02-01    39.49    41.21   37.50     37.61  107419000     23.10015
## 2008-03-01    37.60    39.49   36.10     37.94  130558400     23.30284
## 2008-04-01    38.25    41.00   38.20     39.88  100377800     24.67112
## 2008-05-01    40.10    41.98   39.56     41.31   73131500     25.55578
## 2008-06-01    41.35    41.78   39.71     40.70  103942300     25.17841
## 2008-07-01    40.31    41.44   37.49     38.08  152909100     23.73222
## 2008-08-01    38.34    38.40   36.24     37.66   91668700     23.47046
## 2008-09-01    38.58    38.58   32.51     33.23  155227600     20.70960
## 2008-10-01    33.33    33.41   23.28     28.91  224908800     18.17668
## 2008-11-01    29.46    30.31   25.98     30.12  194890900     18.93745
## 2008-12-01    29.76    29.95   26.99     29.03  152742700     18.25213
##            SMA.SMA10 Cl.gt.SMA Cl.lt.SMA
## 2008-01-01    40.620        NA         1
## 2008-02-01    40.216        NA        NA
## 2008-03-01    39.826        NA        NA
## 2008-04-01    39.856         1        NA
## 2008-05-01    40.187        NA        NA
## 2008-06-01    40.397        NA        NA
## 2008-07-01    40.225        NA         1
## 2008-08-01    39.746        NA        NA
## 2008-09-01    38.796        NA        NA
## 2008-10-01    37.454        NA        NA
## 2008-11-01    36.544        NA        NA
## 2008-12-01    35.686        NA        NA
```

The order_book object: Use the function quantstrat::getOrderBook(.)

```
ob <- getOrderBook(portfolio = portfName)
names(ob)
```

```
## [1] "faberXlu"
```

```
ob$faberXlu$XLU[,1:4]
```

```
##                     Order.Qty Order.Price Order.Type Order.Side
## 2000-04-01 00:00:00 "19000"   "28.484375" "market"   "long"
## 2000-06-01 00:00:00 "all"     "27.25"     "market"   "long"
## 2000-08-01 00:00:00 "19000"   "28.75"     "market"   "long"
## 2001-06-01 00:00:00 "all"     "31.15"     "market"   "long"
## 2003-04-01 00:00:00 "19000"   "19.99"     "market"   "long"
## 2006-03-01 00:00:00 "all"     "30.83"     "market"   "long"
## 2006-06-01 00:00:00 "19000"   "32.290001" "market"   "long"
## 2007-07-01 00:00:00 "all"     "38"        "market"   "long"
## 2007-09-01 00:00:00 "19000"   "39.799999" "market"   "long"
## 2008-01-01 00:00:00 "all"     "39.220001" "market"   "long"
## 2008-04-01 00:00:00 "19000"   "39.880001" "market"   "long"
## 2008-07-01 00:00:00 "all"     "38.080002" "market"   "long"
## 2009-07-01 00:00:00 "19000"   "28.940001" "market"   "long"
## 2010-05-01 00:00:00 "all"     "28.76"     "market"   "long"
## 2010-07-01 00:00:00 "19000"   "30.370001" "market"   "long"
## 2012-11-01 00:00:00 "all"     "35.32"     "market"   "long"
## 2013-01-01 00:00:00 "19000"   "36.580002" "market"   "long"
## 2013-08-01 00:00:00 "all"     "37.299999" "market"   "long"
## 2013-10-01 00:00:00 "19000"   "38.779999" "market"   "long"
## 2013-11-01 00:00:00 "all"     "38.029999" "market"   "long"
## 2014-01-01 00:00:00 "19000"   "39.099998" "market"   "long"
## 2015-03-01 00:00:00 "all"     "44.43"     "market"   "long"
## 2016-01-01 00:00:00 "19000"   "45.419998" "market"   "long"
## 2016-11-01 00:00:00 "all"     "46.75"     "market"   "long"
## 2017-02-01 00:00:00 "19000"   "51.77"     "market"   "long"
## 2017-12-01 00:00:00 "all"     "52.68"     "market"   "long"
```

The `order_book` object (cont'd)

```
ob$faberXlu$XLU[,5:7]
```

```
##                     Order.Threshold Order.Status Order.StatusTime
## 2000-04-01 00:00:00 NA              "closed"     "2000-05-01 00:00:00"
## 2000-06-01 00:00:00 NA              "closed"     "2000-07-01 00:00:00"
## 2000-08-01 00:00:00 NA              "closed"     "2000-09-01 00:00:00"
## 2001-06-01 00:00:00 NA              "closed"     "2001-07-01 00:00:00"
## 2003-04-01 00:00:00 NA              "closed"     "2003-05-01 00:00:00"
## 2006-03-01 00:00:00 NA              "closed"     "2006-04-01 00:00:00"
## 2006-06-01 00:00:00 NA              "closed"     "2006-07-01 00:00:00"
## 2007-07-01 00:00:00 NA              "closed"     "2007-08-01 00:00:00"
## 2007-09-01 00:00:00 NA              "closed"     "2007-10-01 00:00:00"
## 2008-01-01 00:00:00 NA              "closed"     "2008-02-01 00:00:00"
## 2008-04-01 00:00:00 NA              "closed"     "2008-05-01 00:00:00"
## 2008-07-01 00:00:00 NA              "closed"     "2008-08-01 00:00:00"
## 2009-07-01 00:00:00 NA              "closed"     "2009-08-01 00:00:00"
## 2010-05-01 00:00:00 NA              "closed"     "2010-06-01 00:00:00"
## 2010-07-01 00:00:00 NA              "closed"     "2010-08-01 00:00:00"
## 2012-11-01 00:00:00 NA              "closed"     "2012-12-01 00:00:00"
## 2013-01-01 00:00:00 NA              "closed"     "2013-02-01 00:00:00"
## 2013-08-01 00:00:00 NA              "closed"     "2013-09-01 00:00:00"
## 2013-10-01 00:00:00 NA              "closed"     "2013-11-01 00:00:00"
## 2013-11-01 00:00:00 NA              "closed"     "2013-12-01 00:00:00"
## 2014-01-01 00:00:00 NA              "closed"     "2014-02-01 00:00:00"
## 2015-03-01 00:00:00 NA              "closed"     "2015-04-01 00:00:00"
## 2016-01-01 00:00:00 NA              "closed"     "2016-02-01 00:00:00"
## 2016-11-01 00:00:00 NA              "closed"     "2016-12-01 00:00:00"
## 2017-02-01 00:00:00 NA              "closed"     "2017-03-01 00:00:00"
## 2017-12-01 00:00:00 NA              "closed"     "2018-01-01 00:00:00"
```

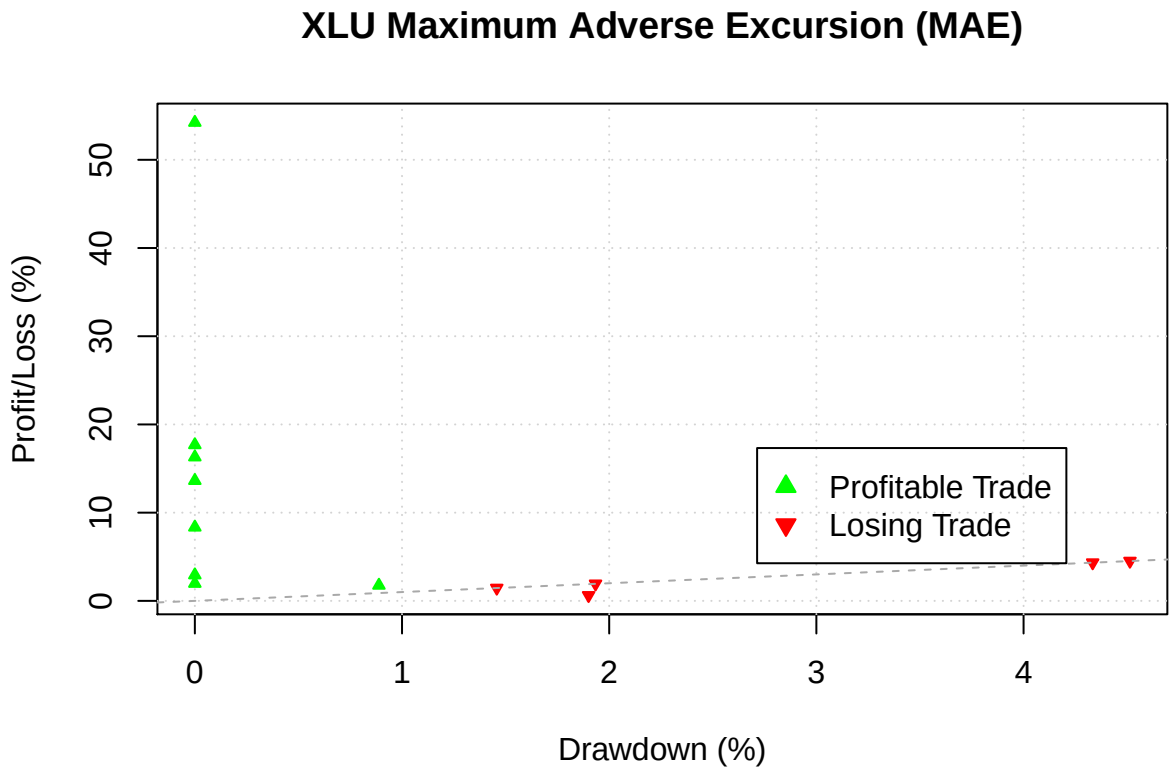
The `order_book` object (cont'd)

```
ob$faberXlu$XLU[,8:11]
```

```
##                     Prefer Order.Set Txn.Fees Rule
## 2000-04-01 00:00:00 ""     NA        "0"      "ruleSignal.rule"
## 2000-06-01 00:00:00 ""     NA        "0"      "ruleSignal.rule"
## 2000-08-01 00:00:00 ""     NA        "0"      "ruleSignal.rule"
## 2001-06-01 00:00:00 ""     NA        "0"      "ruleSignal.rule"
## 2003-04-01 00:00:00 ""     NA        "0"      "ruleSignal.rule"
## 2006-03-01 00:00:00 ""     NA        "0"      "ruleSignal.rule"
## 2006-06-01 00:00:00 ""     NA        "0"      "ruleSignal.rule"
## 2007-07-01 00:00:00 ""     NA        "0"      "ruleSignal.rule"
## 2007-09-01 00:00:00 ""     NA        "0"      "ruleSignal.rule"
## 2008-01-01 00:00:00 ""     NA        "0"      "ruleSignal.rule"
## 2008-04-01 00:00:00 ""     NA        "0"      "ruleSignal.rule"
## 2008-07-01 00:00:00 ""     NA        "0"      "ruleSignal.rule"
## 2009-07-01 00:00:00 ""     NA        "0"      "ruleSignal.rule"
## 2010-05-01 00:00:00 ""     NA        "0"      "ruleSignal.rule"
## 2010-07-01 00:00:00 ""     NA        "0"      "ruleSignal.rule"
## 2012-11-01 00:00:00 ""     NA        "0"      "ruleSignal.rule"
## 2013-01-01 00:00:00 ""     NA        "0"      "ruleSignal.rule"
## 2013-08-01 00:00:00 ""     NA        "0"      "ruleSignal.rule"
## 2013-10-01 00:00:00 ""     NA        "0"      "ruleSignal.rule"
## 2013-11-01 00:00:00 ""     NA        "0"      "ruleSignal.rule"
## 2014-01-01 00:00:00 ""     NA        "0"      "ruleSignal.rule"
## 2015-03-01 00:00:00 ""     NA        "0"      "ruleSignal.rule"
## 2016-01-01 00:00:00 ""     NA        "0"      "ruleSignal.rule"
## 2016-11-01 00:00:00 ""     NA        "0"      "ruleSignal.rule"
## 2017-02-01 00:00:00 ""     NA        "0"      "ruleSignal.rule"
## 2017-12-01 00:00:00 ""     NA        "0"      "ruleSignal.rule"
```

Plot of Maximum Adverse Excursions

```
chart.ME(Portfolio=portfName, Symbol=symbol, type='MAE', scale='percent')
```




```
suppressWarnings(rm.strat(stratName))    # Reset strategy
```

Remarks:

1. We will cover account summary separately, later.

2. There are 33 demo scripts that accompany the quantmod package; these are found under the .../library/quantstrat/demo folder in the package installation. These are good examples to use to get familiar with quantstrat, plus they may be helpful when you work on assignments.

Multiple Fund Example

```
symbols <- c("XLY", "XLP", "XLK")

getSymbols(symbols, index.class=c("POSIXt","POSIXct"),
from=startDate, to=endDate, adjust=TRUE)

for(mySymbol in symbols)
{
  stock(mySymbol, currency="USD", multiplier=1)
  x <- get(mySymbol)
  x <- to.monthly(x, indexAt='endof', drop.time=FALSE)
  colnames(x) <- gsub("x", mySymbol, colnames(x))
  assign(mySymbol, x)
}

# Use same name for strategy, portfolio, account, and orders:
multi.asset <- "multiAsset"
stratName <- multi.asset
portfName <- multi.asset
acctName <- multi.asset

# remove name of the portfolio/account/order book if this is a re-run.
# This allows us to re-initialize the portfolio and account.
rm.strat(portfName)
rm.strat(acctName)

initPortf(multi.asset, symbols=symbols, initDate=initDate)
initAcct(multi.asset, portfolios=multi.asset,
         initDate=initDate, initEq=initEq)
```

Multiple Fund Example: set up quantstrat

```
# Start quantstrat
# Initialize orders and strategy objects
initOrders(portfolio = portfName, initDate = initDate)
strategy(name = stratName, store=TRUE)

# Add an indicator
add.indicator(strategy = stratName, name = "SMA",
              arguments = list(x = quote(Cl(mktdata)), n=10), label="SMA10")

# There are two signals:
# The first is when monthly price crosses over the 10-month SMA
add.signal(strategy = stratName, name = "sigCrossover",
           arguments = list(columns=c("Close","SMA10"),relationship="gte"),
           label="Cl.gt.SMA")

# The second is when the monthly price crosses under the 10-month SMA
add.signal(strategy = stratName, name="sigCrossover",
           arguments = list(columns=c("Close","SMA10"),relationship="lt"),
           label="Cl.lt.SMA")

# There are two rules:
# The first is to buy when the price crosses above the SMA
add.rule(strategy = stratName, name='ruleSignal',
         arguments = list(sigcol="Cl.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 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)
```

Run all three data sets through the same strategy backtest

```
# Run all three data sets through the same strategy backtest
applyStrategy(strategy = stratName, portfolios = portfName)
```

```
## [1] "2000-06-30 00:00:00 XLK 19000 @ 43.5569013754583"
## [1] "2000-07-31 00:00:00 XLK -19000 @ 41.245923909171"
## [1] "2000-08-31 00:00:00 XLK 19000 @ 45.4408503968882"
## [1] "2000-09-29 00:00:00 XLK -19000 @ 37.2770713040254"
## [1] "2002-11-29 00:00:00 XLK 19000 @ 13.7452920603522"
## [1] "2002-12-31 00:00:00 XLK -19000 @ 11.9286225972691"
## [1] "2003-04-30 00:00:00 XLK 19000 @ 12.5814728880656"
## [1] "2004-04-30 00:00:00 XLK -19000 @ 15.738511254395"
## [1] "2004-05-28 00:00:00 XLK 19000 @ 16.3553903110598"
## [1] "2004-07-30 00:00:00 XLK -19000 @ 15.7790969728234"
## [1] "2004-10-29 00:00:00 XLK 19000 @ 16.2011699381321"
## [1] "2005-03-31 00:00:00 XLK -19000 @ 16.1915848150777"
## [1] "2005-05-31 00:00:00 XLK 19000 @ 16.7048158626321"
## [1] "2006-05-31 00:00:00 XLK -19000 @ 17.0276297018167"
## [1] "2006-08-31 00:00:00 XLK 19000 @ 17.6610616940345"
## [1] "2008-01-31 00:00:00 XLK -19000 @ 19.6717267723213"
## [1] "2008-05-30 00:00:00 XLK 19000 @ 21.4543394593653"
## [1] "2008-06-30 00:00:00 XLK -19000 @ 19.4446333891842"
## [1] "2009-05-29 00:00:00 XLK 19000 @ 15.188259313826"
## [1] "2010-06-30 00:00:00 XLK -19000 @ 17.8826933397837"
## [1] "2010-07-30 00:00:00 XLK 19000 @ 19.2151293141205"
## [1] "2010-08-31 00:00:00 XLK -19000 @ 18.1369089746479"
## [1] "2010-09-30 00:00:00 XLK 19000 @ 20.2523330072523"
## [1] "2011-08-31 00:00:00 XLK -19000 @ 21.7385537099868"
## [1] "2011-10-31 00:00:00 XLK 19000 @ 23.2230374903174"
## [1] "2012-10-31 00:00:00 XLK -19000 @ 26.1810761273006"
## [1] "2013-01-31 00:00:00 XLK 19000 @ 26.8150971538177"
```

```
## [1] "2015-08-31 00:00:00 XLK -19000 @ 38.4130347751785"
## [1] "2015-10-30 00:00:00 XLK 19000 @ 41.8664960429116"
## [1] "2016-01-29 00:00:00 XLK -19000 @ 39.7536430449265"
## [1] "2016-03-31 00:00:00 XLK 19000 @ 42.9773561436197"
## [1] "2000-05-31 00:00:00 XLP 19000 @ 15.8983220671391"
## [1] "2001-03-30 00:00:00 XLP -19000 @ 16.6153499430249"
## [1] "2001-11-30 00:00:00 XLP 19000 @ 17.5869993399147"
## [1] "2002-04-30 00:00:00 XLP -19000 @ 17.1077978910717"
## [1] "2003-05-30 00:00:00 XLP 19000 @ 13.9364093587016"
## [1] "2004-07-30 00:00:00 XLP -19000 @ 15.5501045964901"
## [1] "2004-12-31 00:00:00 XLP 19000 @ 16.4875987488728"
## [1] "2007-07-31 00:00:00 XLP -19000 @ 19.7498132112229"
## [1] "2007-08-31 00:00:00 XLP 19000 @ 20.2157517186508"
## [1] "2008-01-31 00:00:00 XLP -19000 @ 20.6605279689954"
## [1] "2008-03-31 00:00:00 XLP 19000 @ 21.2750062627458"
## [1] "2008-06-30 00:00:00 XLP -19000 @ 20.5161118341081"
## [1] "2008-08-29 00:00:00 XLP 19000 @ 21.5821819250725"
## [1] "2008-10-31 00:00:00 XLP -19000 @ 18.6249578629169"
## [1] "2009-06-30 00:00:00 XLP 19000 @ 18.147899695954"
## [1] "2010-06-30 00:00:00 XLP -19000 @ 20.6991407713958"
## [1] "2010-07-30 00:00:00 XLP 19000 @ 21.9005026671474"
## [1] "2010-08-31 00:00:00 XLP -19000 @ 21.4946371618259"
## [1] "2010-09-30 00:00:00 XLP 19000 @ 22.7924585549057"
## [1] "2011-09-30 00:00:00 XLP -19000 @ 24.9481594889317"
## [1] "2011-10-31 00:00:00 XLP 19000 @ 26.1089310682603"
## [1] "2015-08-31 00:00:00 XLP -19000 @ 44.0810843412131"
## [1] "2015-10-30 00:00:00 XLP 19000 @ 46.7755118981612"
## [1] "2016-11-30 00:00:00 XLP -19000 @ 48.615126061916"
## [1] "2017-01-31 00:00:00 XLP 19000 @ 50.9240399624842"
## [1] "2017-10-31 00:00:00 XLP -19000 @ 52.4039824732851"
## [1] "2017-11-30 00:00:00 XLP 19000 @ 55.3257330047777"
## [1] "2018-02-28 00:00:00 XLP -19000 @ 53.1116702727755"
## [1] "1999-10-29 00:00:00 XLY 19000 @ 22.6578228885166"
```

```
## [1] "2000-01-31 00:00:00 XLY -19000 @ 21.7373038716409"
## [1] "2000-03-31 00:00:00 XLY 19000 @ 23.4310114374797"
## [1] "2000-05-31 00:00:00 XLY -19000 @ 21.7046475408582"
## [1] "2001-01-31 00:00:00 XLY 19000 @ 23.03224228914"
## [1] "2001-08-31 00:00:00 XLY -19000 @ 21.3480817781442"
## [1] "2001-11-30 00:00:00 XLY 19000 @ 22.2486094559885"
## [1] "2002-06-28 00:00:00 XLY -19000 @ 22.4092701530507"
## [1] "2003-04-30 00:00:00 XLY 19000 @ 20.8158036596555"
## [1] "2004-07-30 00:00:00 XLY -19000 @ 25.1501588135213"
## [1] "2004-10-29 00:00:00 XLY 19000 @ 26.8672969653228"
## [1] "2005-04-29 00:00:00 XLY -19000 @ 25.8170797085233"
## [1] "2005-05-31 00:00:00 XLY 19000 @ 27.4134998782799"
## [1] "2005-06-30 00:00:00 XLY -19000 @ 27.295324985246"
## [1] "2005-07-29 00:00:00 XLY 19000 @ 28.8441140646038"
## [1] "2005-09-30 00:00:00 XLY -19000 @ 27.1020564725266"
## [1] "2005-11-30 00:00:00 XLY 19000 @ 27.6108944233527"
## [1] "2006-07-31 00:00:00 XLY -19000 @ 27.1277900803625"
## [1] "2006-09-29 00:00:00 XLY 19000 @ 29.4281269883689"
## [1] "2007-07-31 00:00:00 XLY -19000 @ 31.3776787080342"
## [1] "2009-04-30 00:00:00 XLY 19000 @ 20.3624211337708"
## [1] "2010-06-30 00:00:00 XLY -19000 @ 25.9330020094715"
## [1] "2010-07-30 00:00:00 XLY 19000 @ 27.9894829076645"
## [1] "2010-08-31 00:00:00 XLY -19000 @ 26.8766695322902"
## [1] "2010-09-30 00:00:00 XLY 19000 @ 29.8388841617882"
## [1] "2011-08-31 00:00:00 XLY -19000 @ 33.9848083223392"
## [1] "2011-10-31 00:00:00 XLY 19000 @ 35.41739508486"
## [1] "2011-11-30 00:00:00 XLY -19000 @ 35.1632484987767"
## [1] "2011-12-30 00:00:00 XLY 19000 @ 35.622984191041"
## [1] "2015-09-30 00:00:00 XLY -19000 @ 71.534796273366"
## [1] "2015-10-30 00:00:00 XLY 19000 @ 77.9985506301123"
## [1] "2016-01-29 00:00:00 XLY -19000 @ 71.6856204420431"
## [1] "2016-03-31 00:00:00 XLY 19000 @ 76.8273006019278"
```

Run the portfolio, account, and ending equity updates

```
updatePortf(Portfolio = portfName)
```

```
## [1] "multiAsset"
```

```
updateAcct(name = acctName)
```

```
## [1] "multiAsset"
```

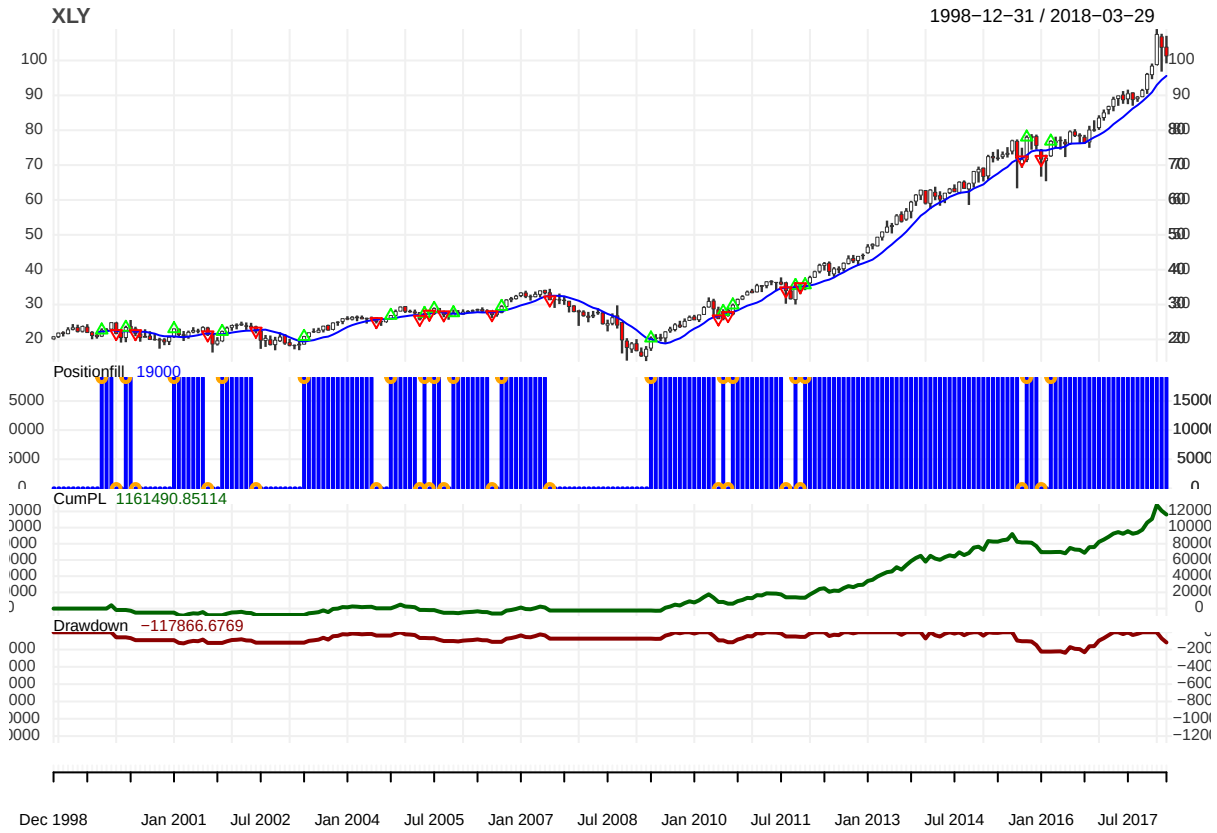
```
updateEndEq(Account = acctName)
```

```
## [1] "multiAsset"
```

Display the summary plots for each ETF

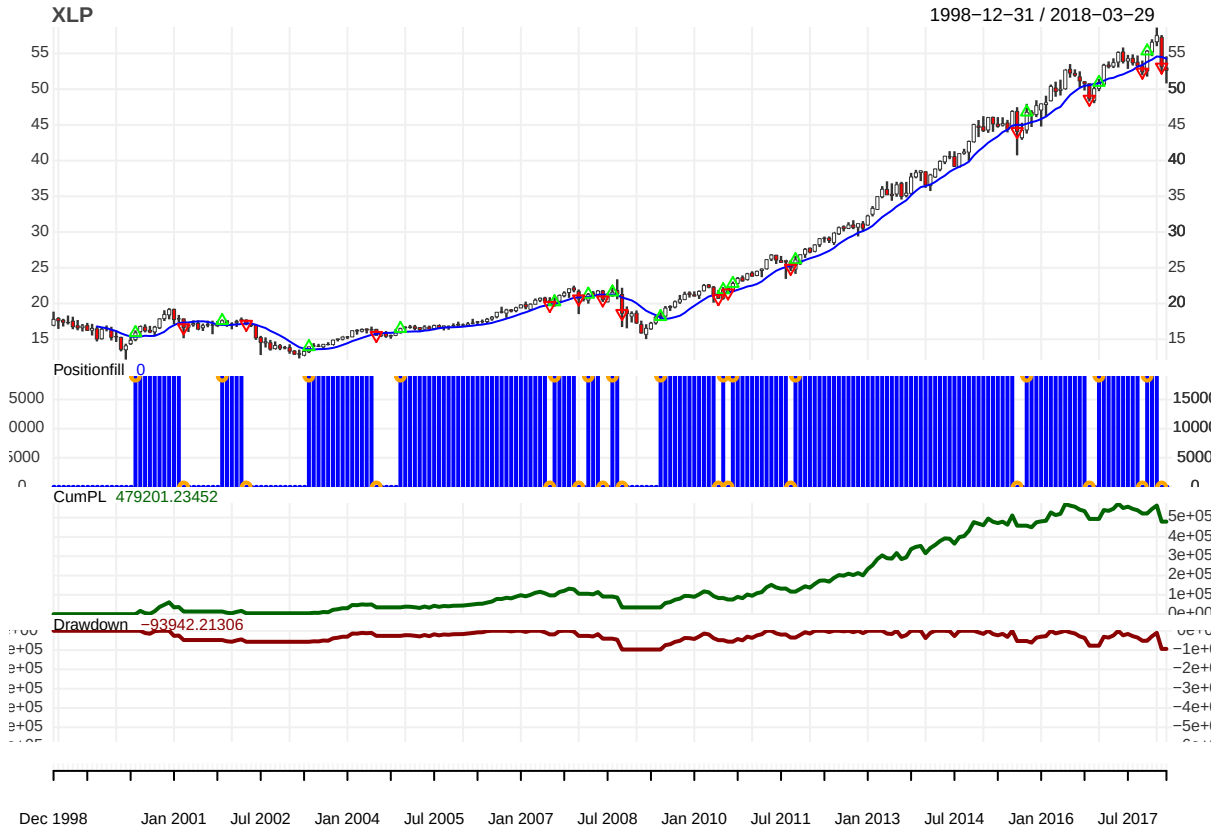
Note: We could loop through these as in the more detailed lecture slides, but they don't get positioned properly using basic R Markdown, so we'll just plot them individually here.

```
chart.Posn(Portfolio = portfName, Symbol = symbols[1],
           TA = "add_SMA(n = 10, col = 'blue')")
```



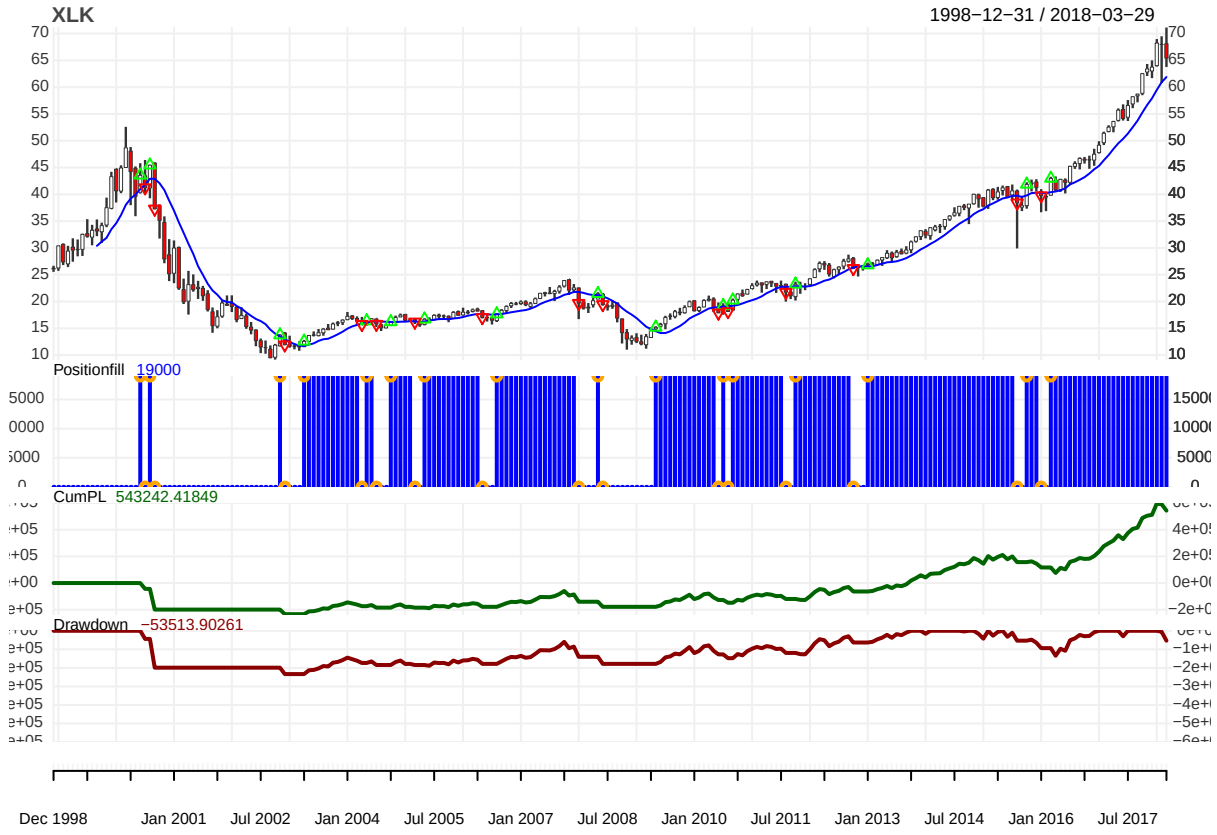
Display the summary plots for each ETF

```
chart.Posn(Portfolio = portfName, Symbol = symbols[2],
           TA = "add_SMA(n = 10, col = 'blue')")
```



Display the summary plots for each ETF

```
chart.Posn(Portfolio = portfName, Symbol = symbols[3],
           TA = "add_SMA(n = 10, col = 'blue')")
```




```
suppressWarnings(rm.strat(stratName))    # Reset strategy
```

Wrap-up

Obtaining the remaining analytics is essentially the same as the single-fund case

More details are available in `CFRM522_014(G)_quantstrat.pdf`