Algorithmic Trading

Netish Joseph

October 5, 2019

Load packages

```
library(foreach)
library(TTR)
library(xts)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(zoo)
library(blotter)
```

```
## Loading required package: FinancialInstrument
```

```
## Loading required package: quantmod
```

```
## Version 0.4-0 included new data defaults. See ?getSymbols.
```

```
## Loading required package: PerformanceAnalytics
```

```
##
## Attaching package: 'PerformanceAnalytics'
```

```
## The following object is masked from 'package:graphics':
##
##     legend
```

```
library(FinancialInstrument)
library(PerformanceAnalytics)
library(quantmod)
library(quantstrat)
library(tseries)
library(roll)
library(pracma)
```

Settings

```
start.pf <- '2012-01-02'
start.date <- '2012-01-01'
end.date <- '2019-10-05'
Sys.setenv(TZ='UTC')
init.eq <- 10000
```

Download Market Data

```
getSymbols(Symbols='AMZN',src='yahoo',from=start.date,to=end.date)
```

```
## 'getSymbols' currently uses auto.assign=TRUE by default, but will
## use auto.assign=FALSE in 0.5-0. You will still be able to use
## 'loadSymbols' to automatically load data. getOption("getSymbols.env")
## and getOption("getSymbols.auto.assign") will still be checked for
## alternate defaults.
##
## This message is shown once per session and may be disabled by setting
## options("getSymbols.warning4.0"=FALSE). See ?getSymbols for details.
```

```
## [1] "AMZN"
```

Instrument Setting

```
currency(primary_id='USD')
```

```
## [1] "USD"
```

```
stock(primary_id='AMZN',currency='USD',multiplier=1)
```

```
## [1] "AMZN"
```

Z score function

```
dAMZN <- diff(Cl(AMZN),lag=1)
dAMZN <- dAMZN[complete.cases(dAMZN)]
adf.test(dAMZN)
```

```
## Warning in adf.test(dAMZN): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  dAMZN
## Dickey-Fuller = -13.188, Lag order = 12, p-value = 0.01
## alternative hypothesis: stationary
```

```
kpss.test(dAMZN)
```

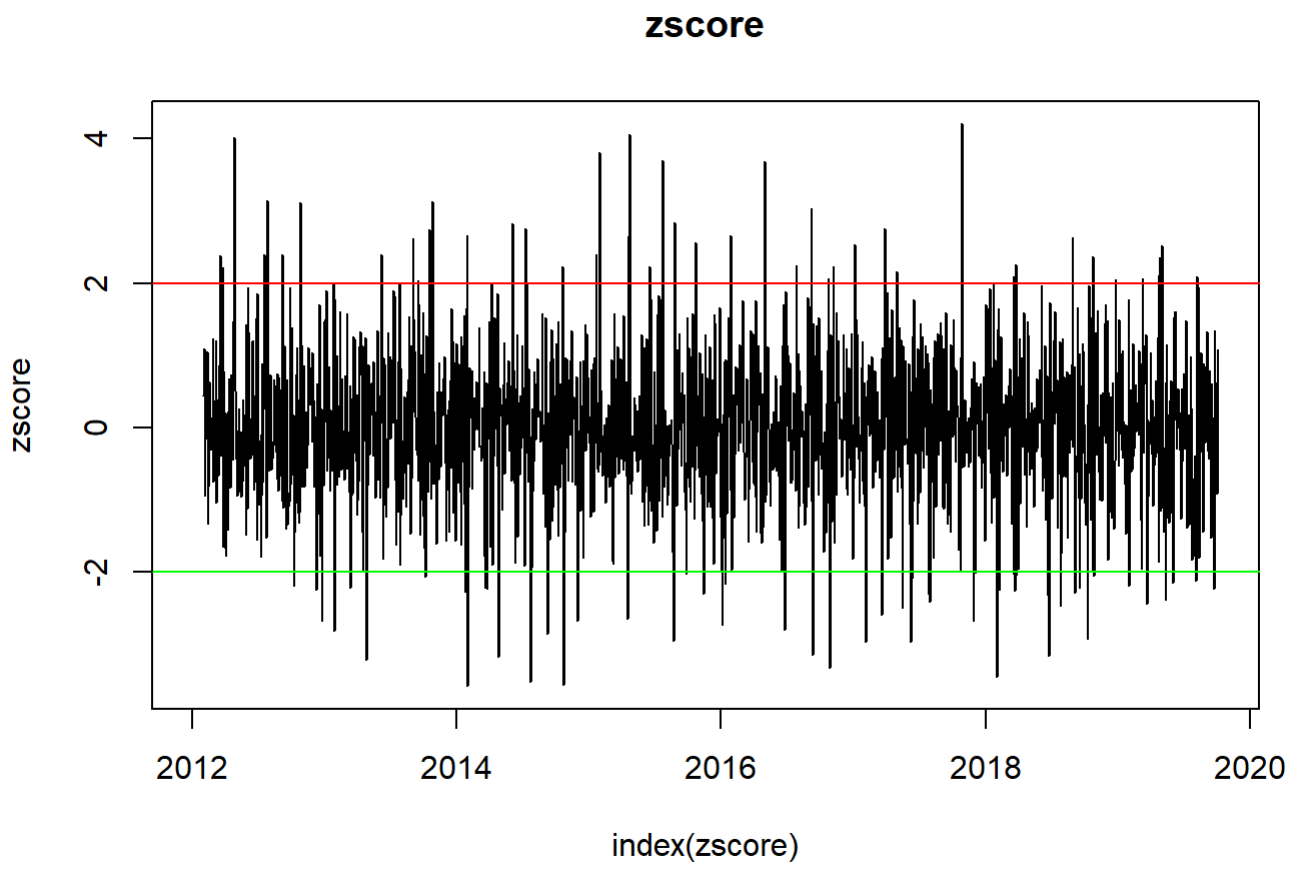
```
## Warning in kpss.test(dAMZN): p-value greater than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  dAMZN
## KPSS Level = 0.15854, Truncation lag parameter = 8, p-value = 0.1
```

```
hurstexp(dAMZN)
```

```
## Simple R/S Hurst estimation:         0.5373018
## Corrected R over S Hurst exponent:   0.5591458
## Empirical Hurst exponent:            0.5260738
## Corrected empirical Hurst exponent:  0.4912963
## Theoretical Hurst exponent:          0.5340438
```

```
zscore.fun <- function(x,n) {roll_scale(x,width=n)}
zscore <- zscore.fun(dAMZN,n=21)
plot(y=zscore,x=index(zscore),type='l',main='zscore')
abline(h=2,col="red")
abline(h=-2,col="green")
```



Create Strategy Objects

```
strat.st <- 'Stratst' # Define Strategy Name
strategy(name=strat.st,store=T)
summary(getStrategy(strat.st))
```

```
##             Length Class  Mode
## name        1      -none- character
## assets      0      -none- NULL
## indicators  0      -none- list
## signals     0      -none- list
## rules       1      -none- list
## constraints 0      -none- NULL
## init        0      -none- list
## wrapup      0      -none- list
## call        3      -none- call
```

```
#rm.strat(strat.st) # Delete existing strategy
```

Initialisation

```
initPortf(name=strat.st,symbols='AMZN',initDate=start.pf)
```

```
## [1] "Stratst"
```

```
initAcct(name=strat.st,portfolios=strat.st,initDate=start.pf,initEq=init.eq)
```

```
## [1] "Stratst"
```

```
initOrders(portfolio=strat.st,initDate=start.pf)
```

SMA Strategy

```
add.indicator(strategy=strat.st,name='SMA',arguments=list(x=quote(Cl(AMZN)),n=20),label='FastSM
A')
```

```
## [1] "Stratst"
```

```
add.indicator(strategy=strat.st,name='SMA',arguments=list(x=quote(Cl(AMZN)),n=50),label='SlowSM
A')
```

```
## [1] "Stratst"
```

```
add.signal(strategy=strat.st,name='sigCrossover',arguments=list(columns=c('FastSMA','SlowSMA'),r
elationship='gt'),label='BuySignal')
```

```
## [1] "Stratst"
```

```
add.signal(strategy=strat.st,name='sigCrossover',arguments=list(columns=c('FastSMA','SlowSMA'),r
elationship='lt'),label='SellSignal')
```

```
## [1] "Stratst"
```

```
addPosLimit(portfolio=strat.st,symbol='AMZN',timestamp=start.date,maxpos=100)
add.rule(strategy=strat.st,name='ruleSignal',arguments=list(sigcol='BuySignal',sigval=T,orderqty
=10,osFUN=osMaxPos,ordertype='market',orderside='long'),type='enter',label='EnterRule',enabled=T
)
```

```
## [1] "Stratst"
```

```
add.rule(strategy=strat.st,name='ruleSignal',arguments=list(sigcol='SellSignal',sigval=T,orderqt
y='all',ordertype='market',orderside='long',TxnFees=-6),type='exit',label="ExitRule",enabled=T)
```

```
## [1] "Stratst"
```

Mean Revision Strategy

```
add.indicator(strategy=strat.st,name='zscore.fun',arguments=list(x=quote(diff(Cl(AMZN),lag=1)),n
=21),label='zscore')
```

```
## [1] "Stratst"
```

```
add.signal(strategy=strat.st,name='sigThreshold',arguments=list(threshold=-2,column='zscore',rel
ationship='lt'),label='BuySignal')
```

```
## [1] "Stratst"
```

```
add.signal(strategy=strat.st,name='sigThreshold',arguments=list(threshold=2,column='zscore',rela
tionship='gt'),label='SellSignal')
```

```
## [1] "Stratst"
```

```
addPosLimit(portfolio=strat.st,symbol='AMZN',timestamp=start.date,maxpos=100)
add.rule(strategy=strat.st,name='ruleSignal',arguments=list(sigcol='BuySignal',sigval=T,orderqty
=10,osFUN=osMaxPos,ordertype='market',orderside='long'),type='enter',label='EnterRule',enabled=T
)
```

```
## [1] "Stratst"
```

```
add.rule(strategy=strat.st,name='ruleSignal',arguments=list(sigcol='SellSignal',sigval=T,orderqt
y='all',ordertype='market',orderside='long',TxnFees=-6),type='exit',label="ExitRule",enabled=T)
```

```
## [1] "Stratst"
```

Strategy Application

```
Mavg.start <- Sys.time()
applyStrategy(strategy=strat.st,portfolios=strat.st)
```

```
## [1] "2012-03-27 00:00:00 AMZN 10 @ 205.440002"
## [1] "2012-10-24 00:00:00 AMZN -10 @ 228.490005"
## [1] "2012-12-12 00:00:00 AMZN 10 @ 251.759995"
## [1] "2013-03-04 00:00:00 AMZN -10 @ 273.109985"
## [1] "2013-03-06 00:00:00 AMZN 10 @ 273.790009"
## [1] "2013-03-20 00:00:00 AMZN -10 @ 257.279999"
## [1] "2013-05-30 00:00:00 AMZN 10 @ 266.829987"
## [1] "2013-08-29 00:00:00 AMZN -10 @ 283.980011"
## [1] "2013-09-25 00:00:00 AMZN 10 @ 312.649994"
## [1] "2014-02-06 00:00:00 AMZN -10 @ 354.589996"
## [1] "2014-06-16 00:00:00 AMZN 10 @ 327.619995"
## [1] "2014-08-14 00:00:00 AMZN -10 @ 333.209991"
## [1] "2014-09-08 00:00:00 AMZN 10 @ 342.339996"
## [1] "2014-10-01 00:00:00 AMZN -10 @ 317.459991"
## [1] "2014-11-28 00:00:00 AMZN 10 @ 338.640015"
## [1] "2014-12-29 00:00:00 AMZN -10 @ 312.040009"
## [1] "2015-02-06 00:00:00 AMZN 10 @ 374.279999"
## [1] "2015-04-14 00:00:00 AMZN -10 @ 385.109985"
## [1] "2015-04-22 00:00:00 AMZN 10 @ 389.799988"
## [1] "2015-09-21 00:00:00 AMZN -10 @ 548.390015"
## [1] "2015-09-22 00:00:00 AMZN 10 @ 538.400024"
## [1] "2016-01-11 00:00:00 AMZN -10 @ 617.73999"
## [1] "2016-03-18 00:00:00 AMZN 10 @ 552.080017"
## [1] "2016-11-11 00:00:00 AMZN -10 @ 739.01001"
## [1] "2017-01-10 00:00:00 AMZN 10 @ 795.900024"
## [1] "2017-08-22 00:00:00 AMZN -10 @ 966.900024"
## [1] "2017-10-17 00:00:00 AMZN 10 @ 1009.130005"
## [1] "2018-04-16 00:00:00 AMZN -10 @ 1441.5"
## [1] "2018-05-08 00:00:00 AMZN 10 @ 1592.390015"
## [1] "2018-10-12 00:00:00 AMZN -10 @ 1788.609985"
## [1] "2019-01-25 00:00:00 AMZN 10 @ 1670.569946"
## [1] "2019-06-04 00:00:00 AMZN -10 @ 1729.560059"
## [1] "2019-07-03 00:00:00 AMZN 10 @ 1939"
## [1] "2019-08-14 00:00:00 AMZN -10 @ 1762.959961"
```

```
Mavg.end <- Sys.time()
Mavg.end-Mavg.start
```

```
## Time difference of 3.049476 secs
```

```
updatePortf(Portfolio=strat.st)
```

```
## [1] "Stratst"
```

```
updateAcct(name=strat.st)
```

```
## [1] "Stratst"
```

```
updateEndEq(Account=strat.st)
```

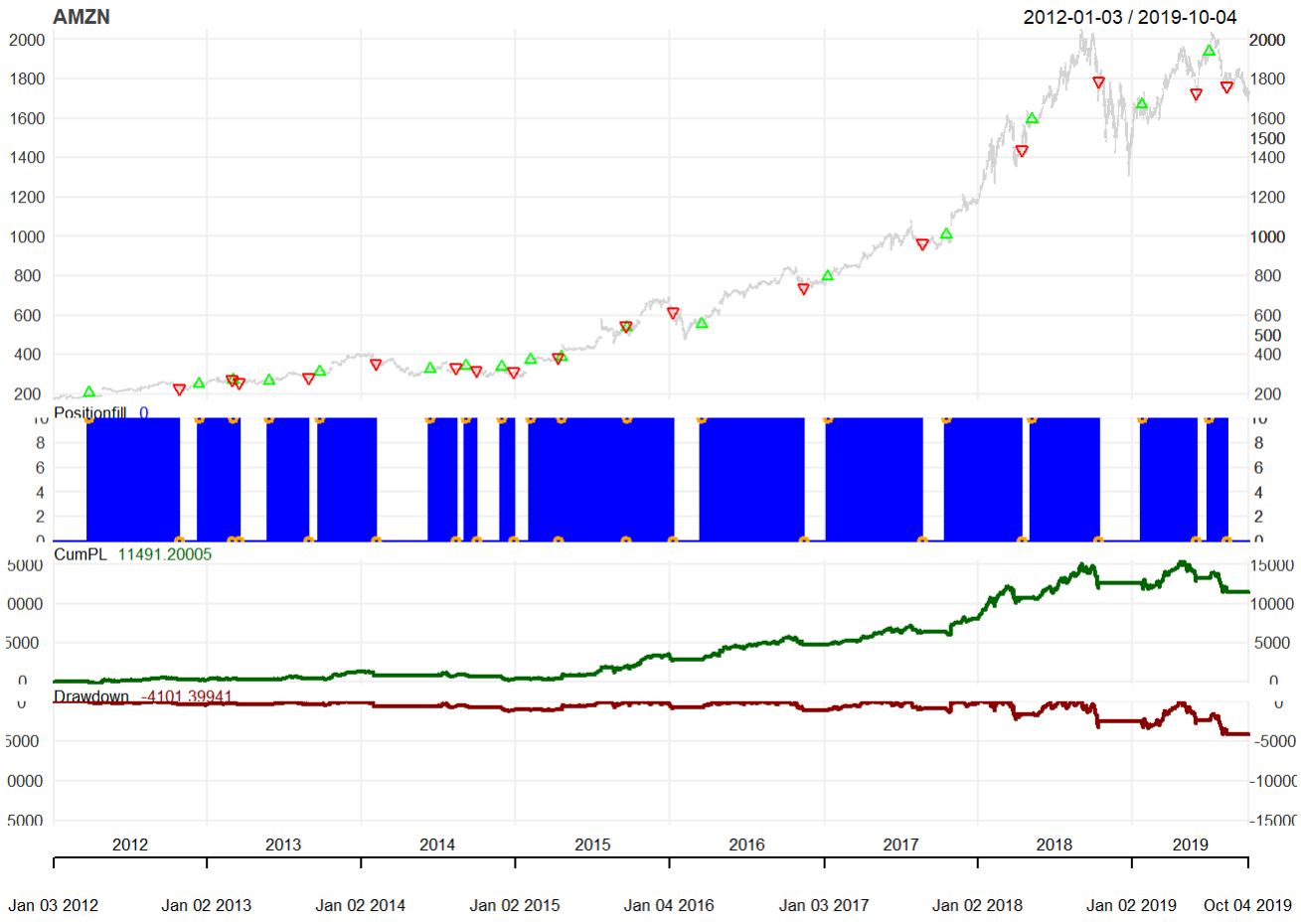
```
## [1] "Stratst"
```

Trading Stats

```
Strat.dstats <- dailyStats(Portfolios=strat.st)
Strat.tstats <- tradeStats(Portfolios=strat.st)
Strat.ptstats <- perTradeStats(Portfolio=strat.st)
```

Position Chart

```
chart.theme <- chart_theme()
chart.theme$col$dn.col <- 'white'
chart.theme$col$dn.border <- 'lightgray'
chart.theme$col$up.border <- 'lightgray'
chart.Posn(Portfolio=strat.st,Symbol='AMZN',theme=chart.theme)
```



```
#getAccount(strat.st)
#tradeStats(Portfolios = strat.st)

final_acct <- getAccount(strat.st)
end_eq <- final_acct$summary$End.Eq
returns <- Return.calculate(end_eq, method="log")
charts.PerformanceSummary(returns, colorset = bluefocus, main = "Strategy Performance")
```