

# Zhang\_Haotian\_FinalProject

Haotian Zhang

5/31/2021

## CFRM522 Final Project: Haotian Zhang

### Strategy Explain

This project will modify the simple fast/slow moving average strategy by considering another financial indicator: Rate of Change. Rate of change is an unbounded financial indicator that can take on both positive and negative values. A positive rate of change indicates an upward trend of the financial instrument, whereas a negative rate of change signifies a downward one. So the idea is that, when we are in long positions only, we want to stay out of the market when there is a downward trend, which is conveniently reflected by Rate of Change.

### Strategy Rules

There are three indicators needed for this strategy, fast moving average, slow moving average, and rate of change. For simplicity's sake, we are in long positions only. The initial equity is set to be 1 million, and we do fixed 2000 shares for each trade. The rate of change period is set to 60 days. The buy/sell rules are listed below:

1. when the fast moving average is greater than the slow moving average, we go long.
2. when the slow moving average is greater than the fast moving average, we exit the market completely.
3. If there is a case where the rate of change indicator is less than -0.1 and we are still in the market because slow moving average has not yet crossed fast moving average, we override the exit rule and exit the market immediately.

### Backtesting

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

```
## Loading required package: xts
```

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

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

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

```
## Loading required package: TTR
```

```
## Registered S3 method overwritten by 'quantmod':  
##   method      from  
##   as.zoo.data.frame zoo
```

```
## Loading required package: blotter
```

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

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

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

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

```
## Loading required package: foreach
```

```
## Loading required package: iterators
```

Let's setup the environment. We will be using VOO as our evaluation stock.

```
symbol = c("VOO")  
currency("USD")  
stock(symbol, currency="USD",multiplier=1)  
Sys.setenv(TZ="UTC")  
initDate = '2010-12-31'  
startDate = '2011-01-01'  
endDate = '2020-12-31'  
getSymbols(symbol,from=startDate,to=endDate,index.class="POSIXct",adjust=T)
```

```
## '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.
```

Let's setup the initial parameters.

```
initEq <- 1e6
fast <- 10
slow <- 100
roc <- 60
```

Let's setup portfolio and strategy to backtest.

```
qs.strategy <- "RateOfChange"
rm.strat(qs.strategy) #for rerun

initPortf(qs.strategy,symbol, initDate=initDate)
initAcct(qs.strategy,portfolios=qs.strategy, initDate=initDate, initEq=initEq)
initOrders(portfolio=qs.strategy,initDate=startDate)
strategy(qs.strategy,store=TRUE)

#indicators
add.indicator(strategy = qs.strategy, name = "SMA",arguments
              =list(x=quote(C1(mktdata)), n=fast), label="fast")
add.indicator(strategy = qs.strategy, name = "SMA",arguments
              =list(x=quote(C1(mktdata)), n=slow), label="slow")
add.indicator(strategy = qs.strategy, name = "ROC", arguments
              =list(x=quote(C1(mktdata)), n=roc), label = "roc")

#signals
add.signal(qs.strategy,name="sigCrossover",arguments
          =list(columns=c("fast","slow"),relationship="gt"),label="fast.gt.slow")
add.signal(qs.strategy,name="sigCrossover",arguments
          =list(columns=c("fast","slow"),relationship="lt"),label="fast.lt.slow")
add.signal(qs.strategy,name = "sigThreshold",arguments =
          list(column="VOO.Close.roc",relationship="lt",
              threshold=-0.1,cross=TRUE),
          label="signal.out")

#rules
add.rule(qs.strategy,
        name='ruleSignal',
        arguments =list(sigcol="fast.gt.slow",sigval=TRUE,
                        replace = TRUE,orderqty = 2000,
                        ordertype='market', orderside='long',
                        orderset='ocolong'),

        type='enter',
        label = 'buy')
add.rule(qs.strategy,
        name='ruleSignal',
        arguments =list(sigcol="fast.lt.slow", sigval=TRUE,
                        replace = TRUE,orderqty='all',
                        ordertype='market',orderside='long',
                        orderset='ocolong'),

        type='exit',
        label = 'sell')
add.rule(qs.strategy, name='ruleSignal',
        arguments = list(sigcol="signal.out", sigval=TRUE,
                        replace = FALSE,
                        orderside='long',
                        ordertype='market',
                        orderqty='all',
                        orderset='ocolong'),

        type='chain', parent = 'buy',
        label = 'override', enabled = FALSE)
```

Run backtest without roc

```
applyStrategy(strategy=qs.strategy , portfolios=qs.strategy)
updatePortf(qs.strategy)
updateAcct(qs.strategy)
updateEndEq(qs.strategy)
```

```
##                VOO
## Net.Trading.PL 75570.5
```

```
##                VOO
## Avg.Trade.PL -3508.926
```

```
##                VOO
## Profit.To.Max.Draw 0.2503075
```

```
##                VOO
## Max.Drawdown -301910.7
```

```
##                VOO
## Ann.Sharpe -1.090164
```

Run backtest with roc

```
##                VOO
## Net.Trading.PL 6669.248
```

```
##                VOO
## Avg.Trade.PL 392.3087
```

```
## VOO
## Profit.To.Max.Draw 0.3778269
```

```
## VOO
## Max.Drawdown -17651.59
```

```
## VOO
## Ann.Sharpe 1.434128
```

The 10/100 day average is chosen randomly, therefore they may not be the best fast/slow averages to consider. In the next section, we will optimize our strategy to find the best fast/slow moving average periods.

## Optimizing parameters

We pick a reasonable range for both fast and slow moving averages, and test for the best pair in this section.

```
initEq <- 1e6
fast <- 10 #dummy
slow <- 100 #dummy
fastRange <- seq(30,55, by=5)
slowRange <- seq(250,325,by=25)
roc <- 60
```

The original strategy has been reinitialized and will not be shown here again. Only the optimization parameters are shown below:

```
add.distribution(strategy = qs.strategy,
  paramset.label = "INDICATOROPT",
  component.type = "indicator",
  component.label = "fast",
  variable = list( n = fastRange ),
  label = "FARange"
)

add.distribution(strategy = qs.strategy,
  paramset.label = "INDICATOROPT",
  component.type = "indicator",
  component.label = "slow",
  variable = list( n = slowRange ),
  label = "SARange"
)
```

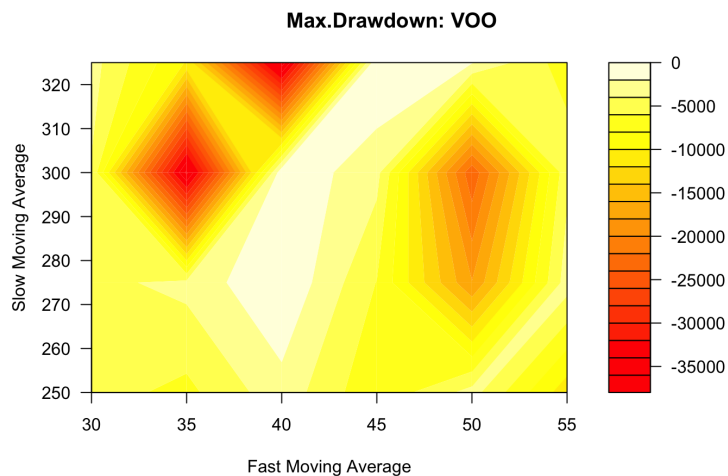
Apply strategy and generate plots for analysis.

```
tradeGraphs(stats = results$tradeStats,
  free.params = c("FARange", "SARange"),
  statistics = c("Profit.To.Max.Draw", "Net.Trading.PL",
    "Max.Drawdown"))
```

```
# Generate heat map (from Guy Yollin's quantstratIII notes).
# FUN can be median or mean.
z <- tapply(X=results$tradeStats$Max.Drawdown,
  INDEX=list(results$tradeStats$FARange,
    results$tradeStats$SARange),
  FUN=mean)

x <- as.numeric(rownames(z))
y <- as.numeric(colnames(z))

filled.contour(x=x,y=y,z=z,color = heat.colors,
  xlab="Fast Moving Average",ylab="Slow Moving Average")
myTitle <- paste0("Max.Drawdown: ", symbol)
title(myTitle)
```



The heat map shows that there is a stable region at 40/275 as the fast/slow moving average pair in Max. Drawdown, which coincides with the 3D plot generated in the previous code chunk (not knitted into the presentation). we will rerun the backtest with this fast/slow pair.

```
##          VOO
## Net.Trading.PL 10692.72
```

```
##          VOO
## Avg.Trade.PL 2138.543
```

```
##          VOO
## Profit.To.Max.Draw 98.85338
```

```
##          VOO
## Max.Drawdown -108.1674
```

```
##          VOO
## Ann.Sharpe 11.76756
```

Performance does seem to improve across the board! Let's try walk forward analysis on max. drawdown to see if our strategy is robust.

## Walk-Forward Analysis

Setup Walk-Forward Analysis

```
qs.strategy <- "RateOfChange"
qs.account <- "opt"
qs.portfolio <- "opt"
suppressWarnings(rm.strat(qs.strategy))
#rm.strat(qs.strategy) #for rerun

strategy(qs.strategy, store=TRUE)

#indicators
add.indicator(strategy = qs.strategy, name = "SMA", arguments
              =list(x=quote(C1(mktdata))), label="fast")
add.indicator(strategy = qs.strategy, name = "SMA", arguments
              =list(x=quote(C1(mktdata))), label="slow")
add.indicator(strategy = qs.strategy, name = "ROC", arguments
              =list(x=quote(C1(mktdata)), n=roc), label = "roc")

#signals
add.signal(qs.strategy, name="sigCrossover", arguments
          =list(columns=c("fast", "slow"), relationship="gt"), label="fast.gt.slow")
add.signal(qs.strategy, name="sigCrossover", arguments
          =list(columns=c("fast", "slow"), relationship="lt"), label="fast.lt.slow")
add.signal(qs.strategy, name = "sigThreshold", arguments =
          list(column="VOO.Close.roc", relationship="lt",
              threshold=-0.1, cross=TRUE),
          label="signal.out")

#rules
add.rule(qs.strategy,
        name='ruleSignal',
        arguments =list(sigcol="fast.gt.slow", sigval=TRUE,
            replace = TRUE, orderqty = 2000,
            ordertype='market', orderside='long',
            orderset='ocolong'),

        type='enter',
        label = 'buy')
add.rule(qs.strategy,
        name='ruleSignal',
        arguments =list(sigcol="fast.lt.slow", sigval=TRUE,
            replace = TRUE, orderqty='all',
            ordertype='market', orderside='long',
            orderset='ocolong'),

        type='exit',
        label = 'sell')
add.rule(qs.strategy, name='ruleSignal',
        arguments = list(sigcol="signal.out", sigval=TRUE,
            replace = FALSE,
            orderside='long',
            ordertype='market',
            orderqty='all',
            orderset='ocolong'),

        type='chain', parent = 'buy',
        label = 'override', enabled = TRUE)

#optimization parameters
add.distribution(strategy = qs.strategy,
                paramset.label = "INDICATOROPT",
                component.type = "indicator",
                component.label = "fast",
                variable = list(n = fastRange ),
                label = "FARange"
)

add.distribution(strategy = qs.strategy,
                paramset.label = "INDICATOROPT",
                component.type = "indicator",
                component.label = "slow",
                variable = list(n = slowRange ),
                label = "SARange"
)

#define WFA
yrsTrain <- 5
yrsTest <- 2
metric <- "Max.Drawdown"
```

```

.obj.func <- function(x)
{
  which(x==max(x) )
}

#initialize portfolio
initPortf(qs.portfolio,symbol, initDate=initDate)
initAcct(qs.account,portfolios=qs.strategy, initDate=initDate, initEq=initEq)
initOrders(portfolio="opt",initDate=startDate)

results <- walk.forward(
  strategy.st=qs.strategy,
  paramset.label='INDICATOROPT',
  portfolio.st="opt",
  account.st="opt",
  period='years',
  k.training=yrsTrain,
  k.testing=yrsTest,
  obj.func=obj.func,
  # obj.args=list(x=quote(tradeStats.list$Net.Trading.PL)),
  obj.args=list(x=quote(tradeStats.list$Max.Drawdown)),
  nsamples=0,
  audit.prefix='wfa',
  anchored=FALSE,
  verbose=TRUE,
  savewf=TRUE,
  saveenv=TRUE # NOTE: Need to set this to TRUE to
               # access the training set results (audit).
)

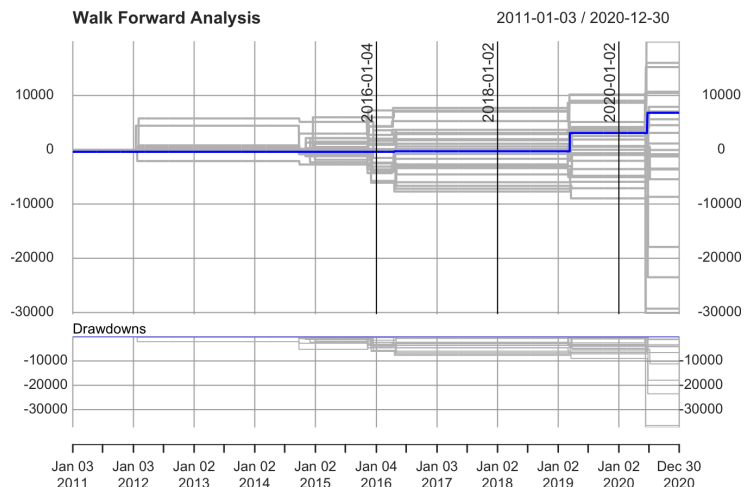
```

```

## Warning in .updatePosPL(Portfolio = pname, Symbol = as.character(symbol), :
## Could not parse //2020-12-30 as ISO8601 string, or one/bothends of the range
## were outside the available prices: 2011-01-03/2020-12-30. Using all data
## instead.

```

Plot WFA



Report WFE

```
## [1] NaN
```

There is no WFE metric because we are out of the market for the most part in this 10-year span, as seen in the walk-forward analysis plot. Therefore our strategy is not as robust on VOO based on Walk-Forward Analysis.

## Monte Carlo Analysis

Let's setup for Monte Carlo Analysis

```

initEq <- 1e6
fast <- 40
slow <- 275
roc <- 60

qs.strategy <- "RateOfChange"
rm.strat(qs.strategy) #for rerun

initPortf(qs.strategy,symbol, initDate=initDate)
initAcct(qs.strategy,portfolios=qs.strategy, initDate=initDate, initEq=initEq)
initOrders(portfolio=qs.strategy,initDate=startDate)
strategy(qs.strategy,store=TRUE)

#indicators
add.indicator(strategy = qs.strategy, name = "SMA",arguments
              =list(x=quote(C1(mktdata)), n=fast), label="fast")
add.indicator(strategy = qs.strategy, name = "SMA",arguments
              =list(x=quote(C1(mktdata)), n=slow), label="slow")
add.indicator(strategy = qs.strategy, name = "ROC", arguments
              =list(x=quote(C1(mktdata)), n=roc), label = "roc")

#signals

```

```

add.signal(qs.strategy,name="sigCrossover",arguments
          =list(columns=c("fast","slow"),relationship="gt"),label="fast.gt.slow")
add.signal(qs.strategy,name="sigCrossover",arguments
          =list(columns=c("fast","slow"),relationship="lt"),label="fast.lt.slow")
add.signal(qs.strategy,name = "sigThreshold",arguments =
          list(column="VOO.Close.roc",relationship="lt",
              threshold=-0.1,cross=TRUE),
          label="signal.out")

#rules
add.rule(qs.strategy,
        name='ruleSignal',
        arguments =list(sigcol="fast.gt.slow",signal=TRUE,
                        replace = TRUE,orderqty = 2000,
                        ordertype='market', orderside='long',
                        orderset='ocolong'),

        type='enter',
        label = 'buy')
add.rule(qs.strategy,
        name='ruleSignal',
        arguments =list(sigcol="fast.lt.slow", signal=TRUE,
                        replace = TRUE,orderqty='all',
                        ordertype='market',orderside='long',
                        orderset='ocolong'),

        type='exit',
        label = 'sell')
add.rule(qs.strategy, name='ruleSignal',
        arguments = list(sigcol="signal.out", signal=TRUE,
                        replace = FALSE,
                        orderside='long',
                        ordertype='market',
                        orderqty='all',
                        orderset='ocolong'),

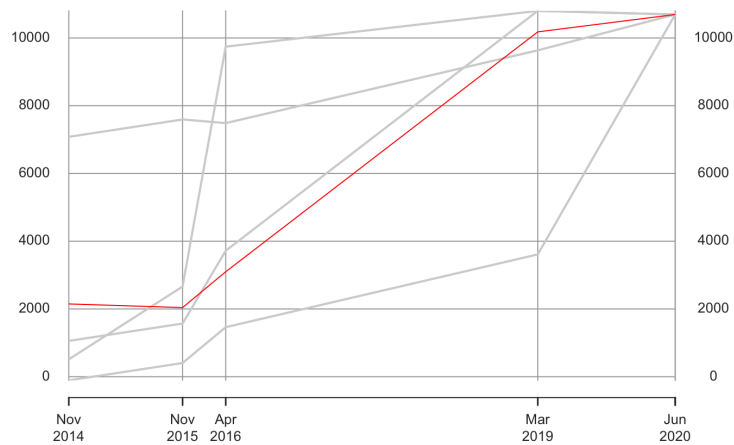
        type='chain', parent = 'buy',
        label = 'override', enabled = TRUE)

applyStrategy(strategy=qs.strategy , portfolios=qs.strategy)
updatePortf(qs.strategy)
updateAcct(qs.strategy)
updateEndEq(qs.strategy)

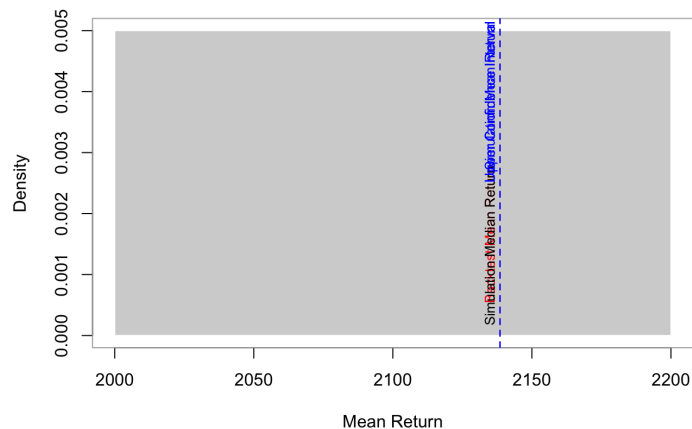
```

MCSIM without replacement

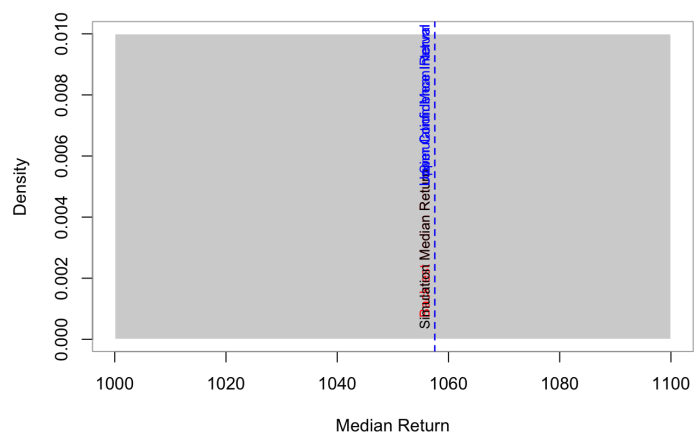
4 replicates without replacement and block length 1 2014-11-05 / 2020-06-18



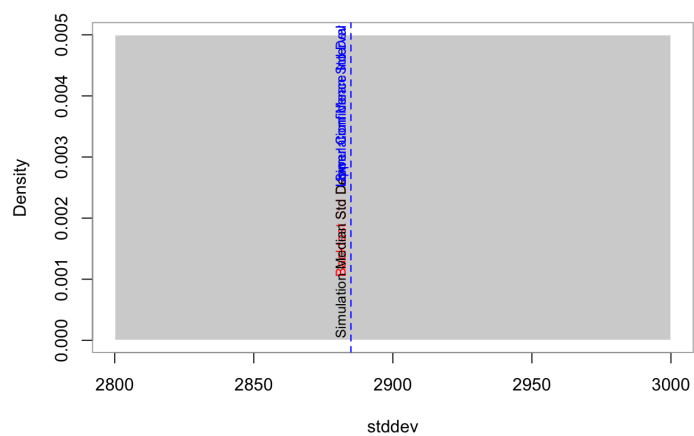
Mean distribution of 4 replicates without replacement using block length 1 and 0.95 confidence interval



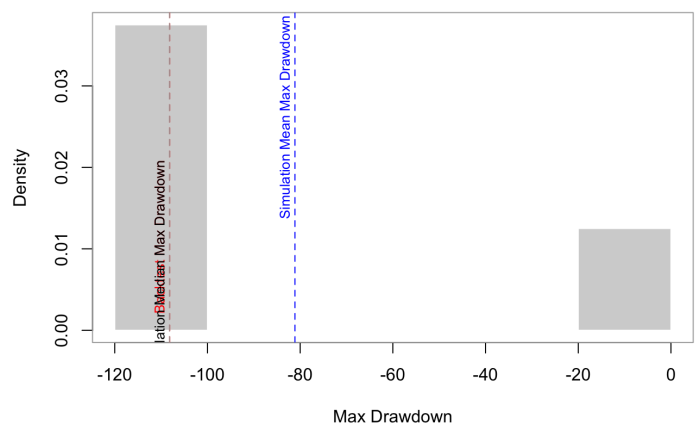
Median distribution of 4 replicates without replacement using block length 1 and 0.95 confidence interval



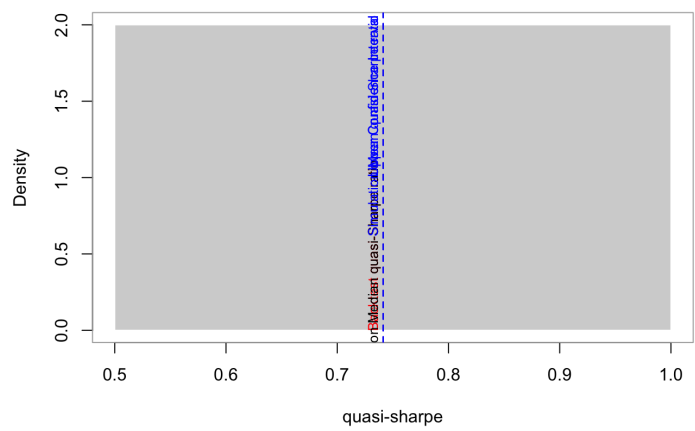
Std Dev distribution of 4 replicates without replacement using block length 1 and 0.95 confidence interval



maxDrawdown distribution of 4 replicates without replacement using block length 1 and 0.95 confidence interval



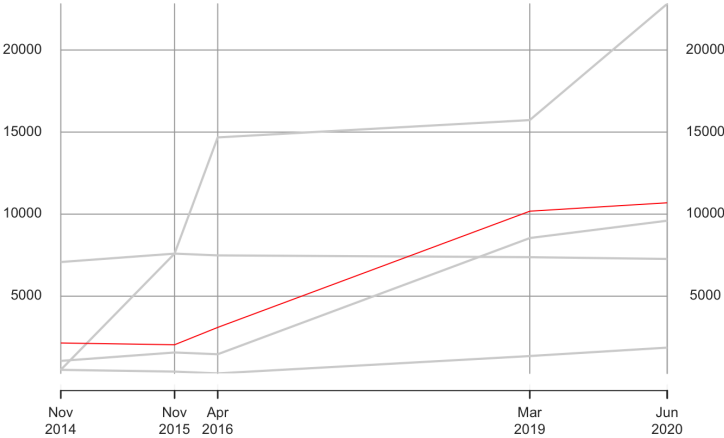
quasi-Sharpe distribution of 4 replicates without replacement using block length 1 and 0.95 confidence interval



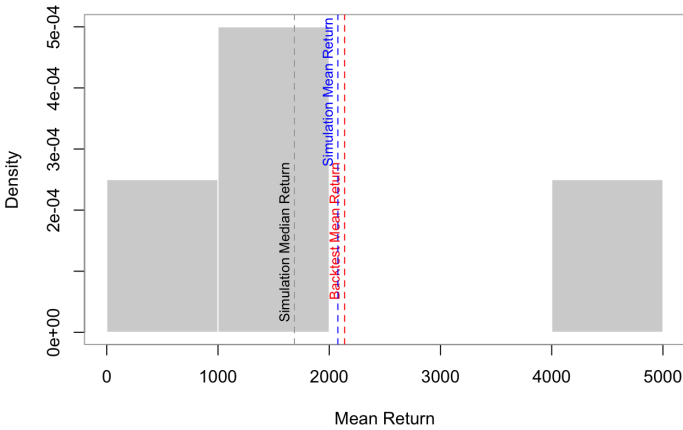
MCSIM with replacement



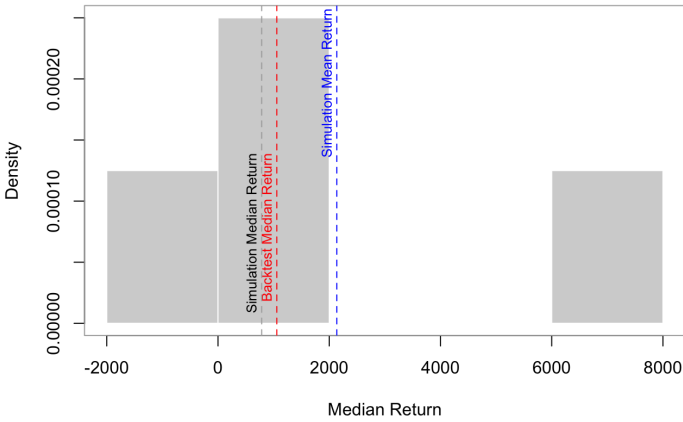
4 replicates with replacement and block length 1 2014-11-05 / 2020-06-18

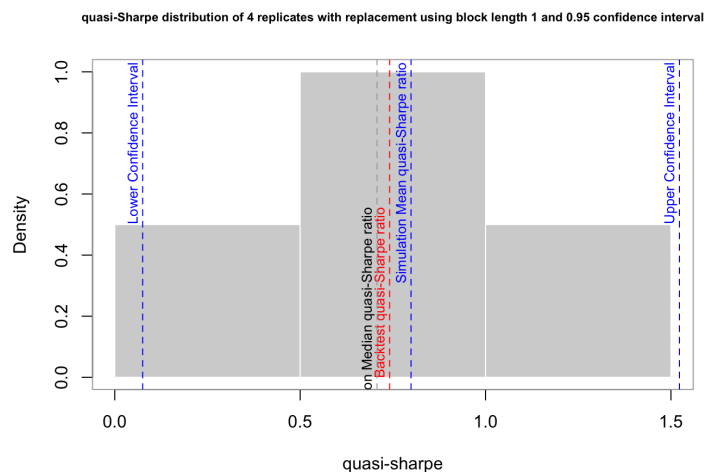
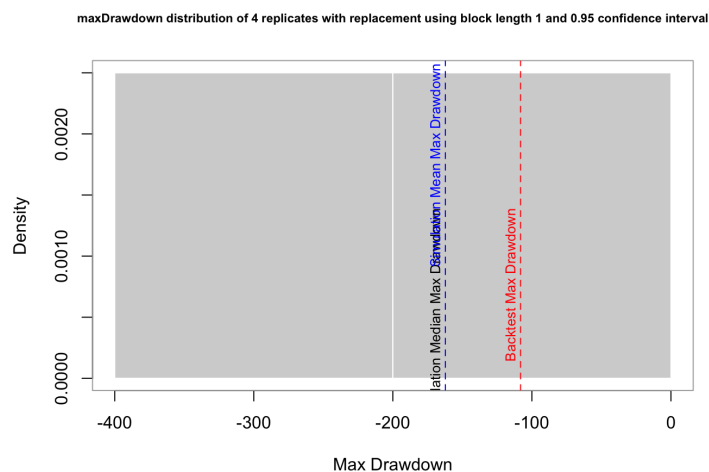
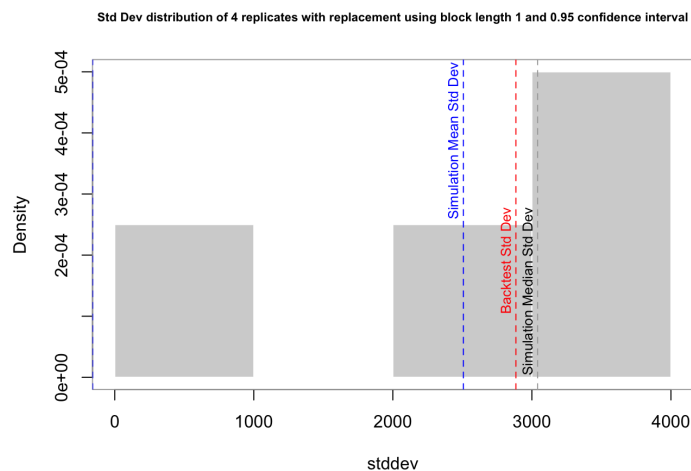


Mean distribution of 4 replicates with replacement using block length 1 and 0.95 confidence interval



Median distribution of 4 replicates with replacement using block length 1 and 0.95 confidence interval





Here is a summary of the CI's for different performance metrics

	Sample Mean	Sample Median	Backtest	Lower CI	Upper CI	Std. Error
## mean	1.20460989	0.88325562	0.56058797	-0.90233731	3.3115571	1.0749928
## median	0.04549036	0.03621899	0.05039319	-0.08097534	0.1719561	0.0645245
## stddev	2.57305215	1.86704115	0.99029995	-2.33101162	7.4771159	2.5021193
## maxDD	-0.13310120	-0.05577504	-0.05031353	-0.51318164	0.2469792	0.1939222
## sharpe	0.53972914	0.47709146	0.99029995	0.22060960	0.8588487	0.1628191

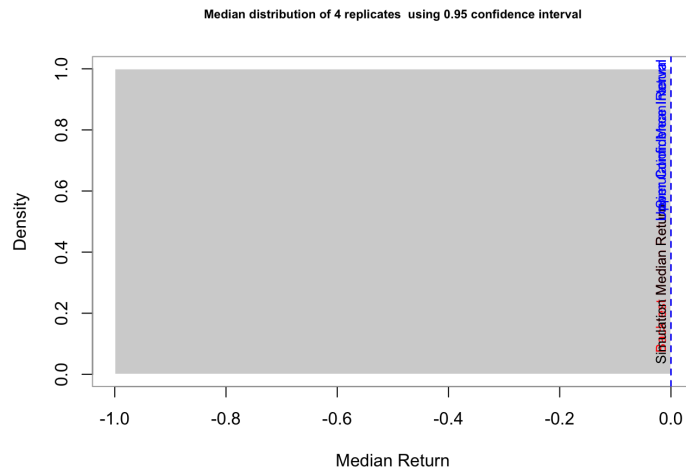
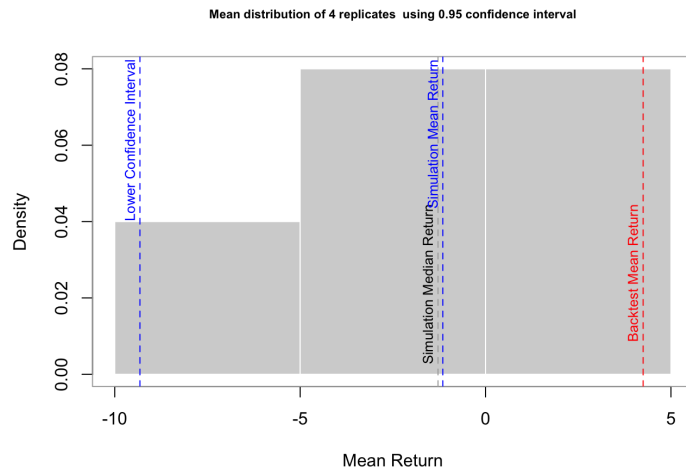
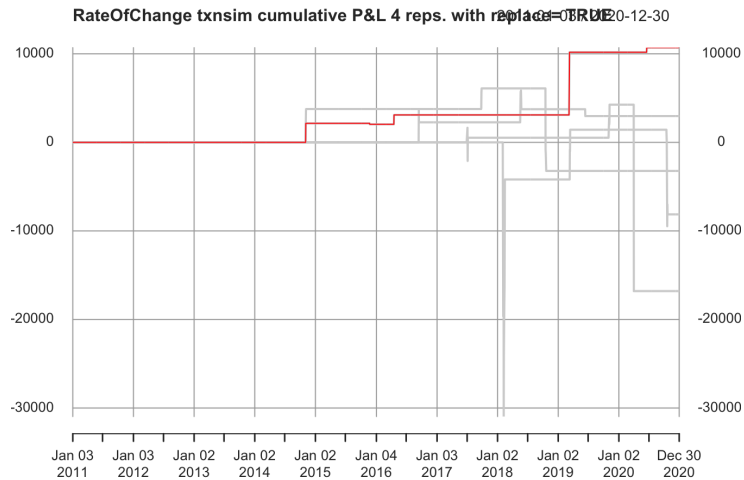
TXNSIM without replacement

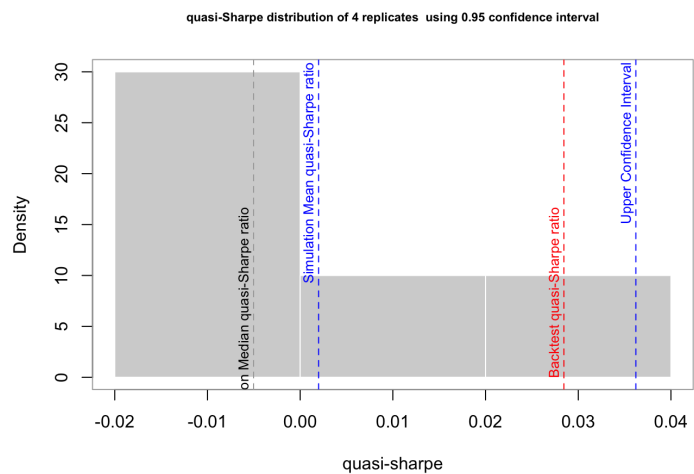
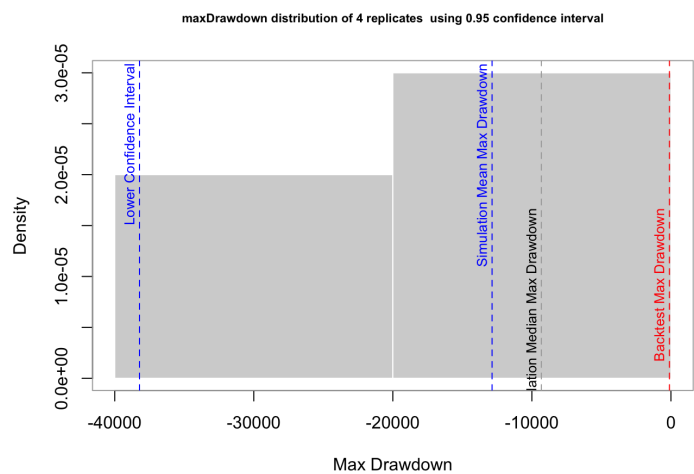
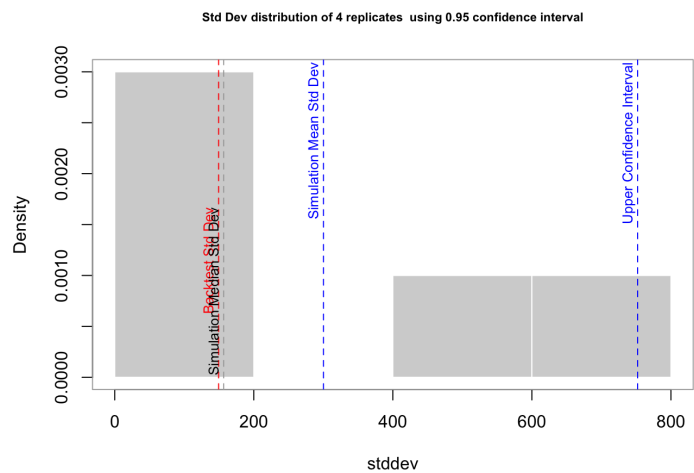
```
## Warning in if (class(index(prices)) != "POSIXct" | class(index(prices)) != : the
## condition has length > 1 and only the first element will be used

## Warning in if (class(index(prices)) != "POSIXct" | class(index(prices)) != : the
## condition has length > 1 and only the first element will be used

## Warning in if (class(index(prices)) != "POSIXct" | class(index(prices)) != : the
## condition has length > 1 and only the first element will be used
```

```
## Warning in if (class(index(prices)) != "POSIXct" | class(index(prices)) != : the
## condition has length > 1 and only the first element will be used
```





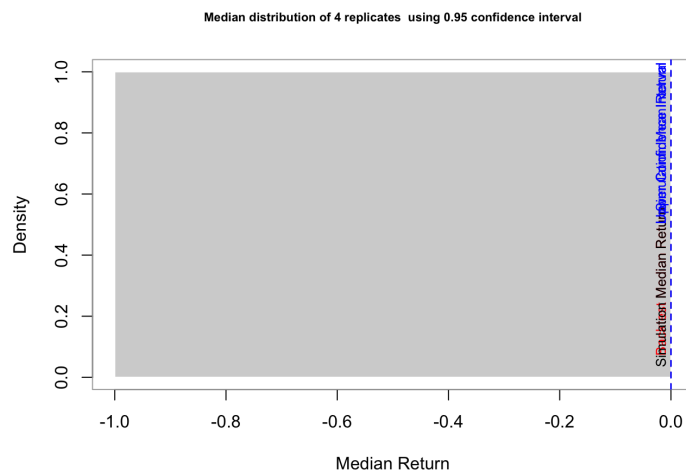
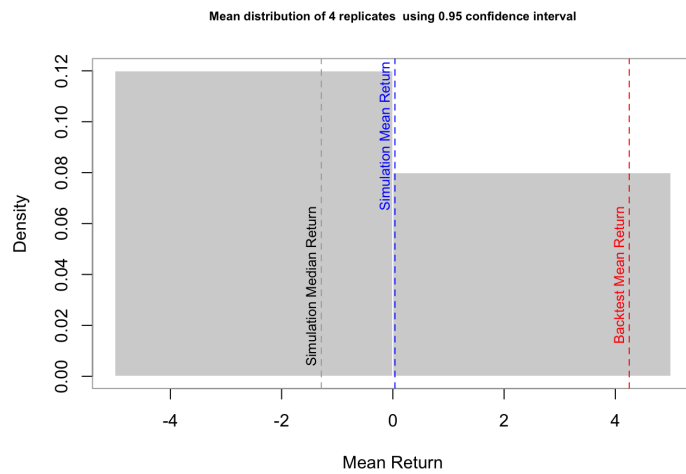
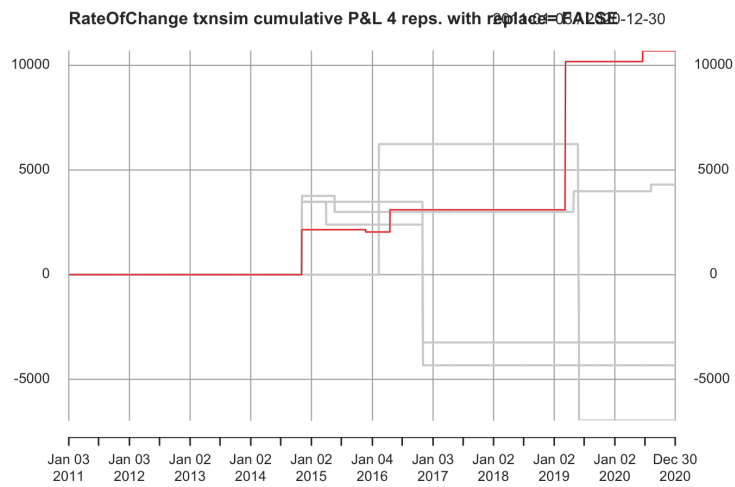
TXNSIM with replacement

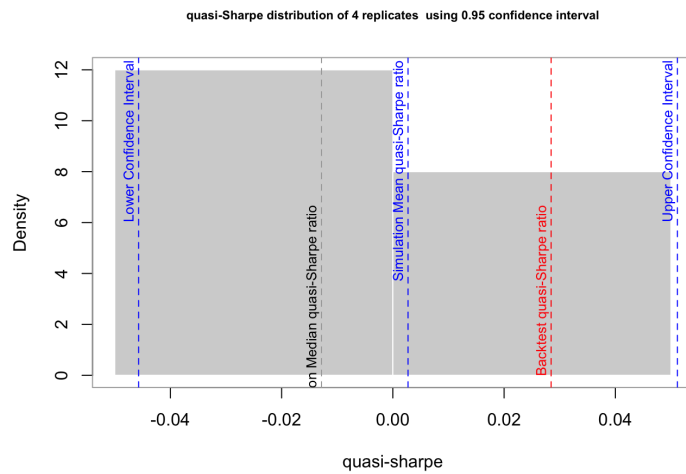
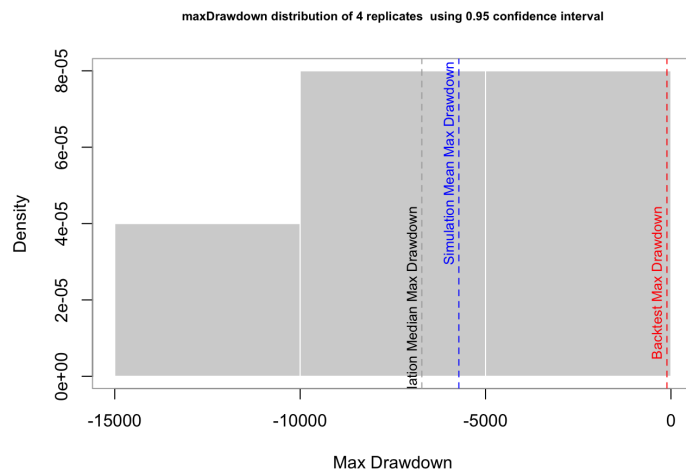
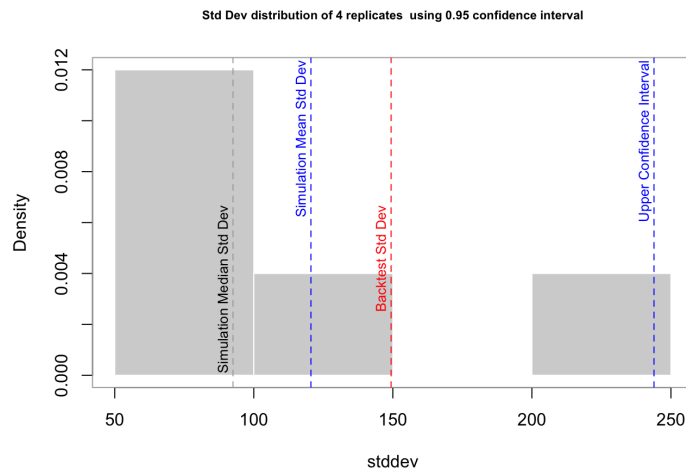
```
## Warning in if (class(index(prices)) != "POSIXct" | class(index(prices)) != : the
## condition has length > 1 and only the first element will be used

## Warning in if (class(index(prices)) != "POSIXct" | class(index(prices)) != : the
## condition has length > 1 and only the first element will be used

## Warning in if (class(index(prices)) != "POSIXct" | class(index(prices)) != : the
## condition has length > 1 and only the first element will be used

## Warning in if (class(index(prices)) != "POSIXct" | class(index(prices)) != : the
## condition has length > 1 and only the first element will be used
```





Here shows the quantiles of cumulative P&L

##	0%	25%	50%	75%	100%
##	-6958.792	0.000	0.000	3098.249	10692.715

Here is a summary of the CI's for different performance metrics

##	backtest	Std. Error	Lower CI	Upper CI
## mean	4.249887e+00	2.8827136	-5.612531e+00	5.687498e+00
## median	0.000000e+00	0.0000000	0.000000e+00	0.000000e+00
## stddev	1.493686e+02	62.9386370	-2.811998e+00	2.439029e+02
## maxDD	-1.081674e+02	5414.9799215	-1.633297e+04	4.893357e+03
## sharpe	2.845234e-02	0.0247106	-4.570513e-02	5.115862e-02
## totalPL	1.069272e+04	7252.9074312	-1.412113e+04	1.430975e+04

## Conclusion

This modified moving average strategy does not perform well on VOO etf. The optimum parameters show that we are out of the market for the most time. Therefore, this is not a viable strategy for VOO and more deliberations are needed before apply this strategy.