

This will be a momentum strategy. Much of this will come from Ross Bennett's blog.

1 Preparation

```
require(quantstrat)
currency("USD")

## [1] "USD"

symbols = c("XLY", "XLP", "XLE", "XLF")
stock(symbols, currency = "USD", multiplier = 1)

## [1] "XLY" "XLP" "XLE" "XLF"

getSymbols(symbols, src = "yahoo", index.class = c("POSIXt", "POSIXct"), from = "2000-01-01")

## [1] "XLY" "XLP" "XLE" "XLF"
```

Convert to monthly and drop all columns except Adjusted Close

```
for (symbol in symbols) {
  x <- get(symbol)
  x <- to.monthly(x, indexAt = "lastof", drop.time = TRUE)
  indexFormat(x) <- "%Y-%m-%d"
  colnames(x) <- gsub("x", symbol, colnames(x))
  x <- x[, 6] #drops all columns except Adjusted Close which is 6th column
  assign(symbol, x)
}
```

There are now four objects that have an adjusted close. Merge the symbols into a single object with just the close prices

```
symbols_close <- do.call(merge, lapply(symbols, get))
head(symbols_close)
```

	XLY.Adjusted	XLP.Adjusted	XLE.Adjusted	XLF.Adjusted
## 2000-01-31	23.75	17.94	22.57	17.84
## 2000-02-29	22.43	15.84	21.62	15.93
## 2000-03-29	25.56	16.39	24.22	18.78
## 2000-04-28	25.03	17.25	23.86	18.96
## 2000-05-29	23.67	18.49	26.67	19.38
## 2000-06-28	22.39	19.53	25.19	18.44

Now calculate the three period rate of change so that the returns can be ranked to see where the momentum lies.

```
roc <- ROC(symbols_close, n = 3, type = "discrete")
head(roc)
```

##	XLY.Adjusted	XLP.Adjusted	XLE.Adjusted	XLF.Adjusted
## 2000-01-31	NA	NA	NA	NA
## 2000-02-29	NA	NA	NA	NA
## 2000-03-29	NA	NA	NA	NA
## 2000-04-28	0.05389	-0.03846	0.05716	0.06278
## 2000-05-29	0.05528	0.16730	0.23358	0.21657
## 2000-06-28	-0.12402	0.19158	0.04005	-0.01810

Now apply the rank function across each column. The symbol with the highest return has the rank one for each month. This will be an xts object with ranks.

```
r <- as.xts(t(apply(-roc, 1, rank)))
head(r)
```

##	XLY.Adjusted	XLP.Adjusted	XLE.Adjusted	XLF.Adjusted
## 2000-01-31	1	2	3	4
## 2000-02-29	1	2	3	4
## 2000-03-29	1	2	3	4
## 2000-04-28	3	4	2	1
## 2000-05-29	4	3	1	2
## 2000-06-28	4	1	2	3

2 Functions

2.1 RankingRB

Computes the rank of an xts object of ranking factors. Ranking factors are the factors that are ranked (i.e. asset returns)

- args x = xts object of ranking factors
- Returns Returns an xts object with ranks (e.g. for ranking asset returns, the asset with the greatest return receives a rank of 1)

```
RankRB <- function(x) {
  r <- as.xts(t(apply(-x, 1, rank, na.last = "keep")))
  return(r)
}
```

2.2 MontlyAd

Converts daily data to monthly and returns only the monthly close.

- Args: x = daily price data from Yahoo Finance
- Returns xts object with the monthly adjusted close prices

```
MonthlyAd <- function(x) {  
  sym <- sub("\\\\.*$", "", names(x)[1])  
  Ad(to.monthly(x, indexAt = "lastof", drop.time = TRUE, name = sym))  
}
```

2.3 CAGR

Function to compute the CAGR given simple returns

- Args: x = xts of simple returns m = periods per year (i.e. monthly = 12, daily = 252)
- Returns the Compound Annual Growth Rate

```
CAGR <- function(x, m) {  
  x <- na.omit(x)  
  cagr <- apply(x, 2, function(x, m) prod(1 + x)^(1/(length(x)/m)) - 1, m = m)  
  return(cagr)  
}
```

2.4 SimpleMomentumTest

Returns a list containing a matrix of individual asset returns and the combined returns. Trade the top n asset(s) if the rank of last period is less than or equal to n, then I would experience the return for this month.

- args: xts.ret = xts of one period returns xts.rank = xts of ranks n = number of top ranked assets to trade ret.fill.na = number of return periods to fill with NA
- Returns: An xts object of simple returns

```
SimpleMomentumTest <- function(xts.ret, xts.rank, n = 1, ret.fill.na = 3) {  
  # returns a list containing a matrix of individual asset returns and the  
  # combined returns args: xts.ret = xts of one period returns xts.rank =  
  # xts of ranks n = number of top ranked assets to trade ret.fill.na =
```

```

# number of return periods to fill with NA
#
# Returns: returns an xts object of simple returns

# trade the top n asset(s) if the rank of last period is less than or
# equal to n, then I would experience the return for this month.

# lag the rank object by one period to avoid look ahead bias
lag.rank <- lag(xts.rank, k = 1, na.pad = TRUE)
n2 <- nrow(lag.rank[is.na(lag.rank[, 1]) == TRUE])
z <- max(n2, ret.fill.na)

# for trading the top ranked asset, replace all ranks above n with NA to
# set up for element wise multiplication to get the realized returns
lag.rank <- as.matrix(lag.rank)
lag.rank[lag.rank > n] <- NA
# set the element to 1 for assets ranked <= to rank
lag.rank[lag.rank <= n] <- 1

# element wise multiplication of the 1 period return matrix and lagged
# rank matrix
mat.ret <- as.matrix(xts.ret) * lag.rank

# average the rows of the mat.ret to get the return for that period
vec.ret <- rowMeans(mat.ret, na.rm = TRUE)
vec.ret[1:z] <- NA

# convert to an xts object
vec.ret <- xts(x = vec.ret, order.by = index(xts.ret))
f <- list(mat = mat.ret, ret = vec.ret, rank = lag.rank)
return(f)
}

```