

Composite Dual Momentum, Replication

matReturns.com

```
require(quantmod)
require(PerformanceAnalytics)

##### Replication of the Composite Dual Momentum- Gary Antonacci #####

### By: matReturns.com
#
# In this illustration, I replicate Composite Dual Momentum from
# Gary Antonacci's 2015 white paper "Risk Premia Harvesting Through Dual Momentum"
#
# Composite Dual Momentum splits the portfolio into four equal quarters.
# Each quarter has two assets that get measured against each other.
# The cumulative returns from the last 12 months are measured.
# The asset with the highest cum ret is selected then measured against the proxy.
# If cum ret from asset is greater than proxy go long asset, else go to cash.
#
#
#

#####
composite_dual_momentum <- function(startDate = '2000-01-01',
                                     endDate = '2023-12-31',
                                     period = "monthly",
                                     look_back = 12) {
  # Define the ETF tickers for the 9 global asset classes
  tickers <- c(
    "SPY", # SPDR S&P 500 ETF Trust
    "EFA", # iShares MSCI EAFE ETF
    "HYG", # iShares iBoxx $ High Yield Corporate Bond ETF
    "BIV", # Vanguard Intermediate-Term Bond Index Fund
    "VNQ", # Vanguard Real Estate Index ETF
    "REM", # iShares Mortgage Real Estate Capped ETF
    "GLD", # SPDR Gold Shares
    "TLT", # iShares 20+ Year Treasury Bond ETF
    "BIL" # SPDR Bloomberg Barclays 1-3 Month T-Bill ETF
  )

  # Get adjusted prices of tickers from 1-1-2000, column bind the data
  prices <- list()
  for (i in 1:length(tickers)) {
```

```

ticker <- Ad(get(getSymbols(tickers[i], from = '2000-01-01'))))
colnames(ticker) <- tickers[i]
prices[[i]] <- ticker
}
prices <- na.omit(do.call(cbind, prices))

# Create price and return data based on the specified period
if (tolower(period) == "daily") {
  period.prices <- prices
  period.rets <- na.omit(Return.calculate(prices))
} else if (tolower(period) == "weekly") {
  period.prices <- to.period(prices, period = "weeks", OHLC = FALSE)
  period.rets <- na.omit(Return.calculate(period.prices))
} else if (tolower(period) == "monthly") {
  period.prices <- to.period(prices, period = "months", OHLC = FALSE)
  period.rets <- na.omit(Return.calculate(period.prices))
} else {
  stop("Invalid period. Please use 'daily', 'weekly', or 'monthly'.")
}

# Initialize empty lists to store the results
assets_lists <- vector("list", length(tickers))
momsig_lists <- vector("list", length(tickers))

# Loop through length of return column less the look_back for momentum calculation
for (i in 1:(nrow(period.rets) - look_back)) {
  # Calculate the look_back period cumulative returns
  look_back_returns <- period.rets[c(i:(i + look_back - 1)), ]
  momentums <- Return.cumulative(look_back_returns)

  for (j in seq(from = 1, to = (length(tickers) - 1), by = 2)) {
    # Set conditions for asset selection
    # Compare momentum of assets and asset to proxy
    # Store return in respective list
    if (momentums[, j] > momentums[, j + 1]) {
      assets_lists[[j]][i] <- period.rets[i + look_back, j]
      momsig_lists[[j]][i] <- momentums[, j]
    } else {
      assets_lists[[j]][i] <- period.rets[i + look_back, j + 1]
      momsig_lists[[j]][i] <- momentums[, j + 1]
    }

    if (momsig_lists[[j]][i] > momentums[, length(tickers)]) {
      assets_lists[[j]][i] <- assets_lists[[j]][i]
    } else {
      assets_lists[[j]][i] <- 0
    }
  }
}

# Row bind the list from all the modules
assets_lists <- do.call(cbind, assets_lists)

```

```

# Sum rows(returns) of the modules and divide by the number of modules to get CDM strategy returns,
CDM_strat <- list()
for (i in 1:nrow(assets_lists)) {
  CDM_strat[[i]] <- sum(as.numeric(c(assets_lists[i, ])) / ncol(assets_lists))
}
CDM_strat <- do.call(rbind, CDM_strat)
colnames(CDM_strat) <- "CDM"

# Put all the data in the same dataframe
# Merge the period prices (excluding the first 'look_back' rows) with the CDM strategy returns column
CDM_XTS <- cbind(period.prices[(look_back + 1):(nrow(period.prices) - 1)], CDM_strat)

# Calculate the returns for SPY and TLT
spyRets <- Return.calculate(CDM_XTS$SPY)
tlRets <- Return.calculate(CDM_XTS$TLT)

# Create a benchmark portfolio with 60% SPY and 40% TLT weights, rebalanced annually
benchmark <- Return.portfolio(R = cbind(spyRets, tlRets), weights = c(0.60, 0.40), rebalance_on = 1)

# Combine the CDM strategy returns and benchmark returns into a new data frame
CDM_test <- cbind(CDM_XTS$CDM, benchmark)
colnames(CDM_test) <- c("CDM", "Benchmark.60/40")

# Return the final data frame containing CDM strategy returns and benchmark returns
return(CDM_test)
}

startDate = '2000-01-01'
endDate = '2023-12-31'
period = "monthly"
look_back = 12

# Calculate CDM strategy returns for monthly data with a look_back of 20
CDM <- composite_dual_momentum(startDate, endDate, "monthly", look_back)

```

Thanks for clicking by, please check out matReturns.com for more.

Creative Commons License

This work is licensed under a Creative Commons Attribution 4.0 International License.

Note: The code and data provided in this analysis are for illustrative purposes only and do not constitute financial advice. Investors should conduct thorough due diligence before making any investment decisions.