Hedge Fund Strategy - Momentum + Value

2023-01-08

Since the current market shows high volatility and low predictability, here we choose to apply some modification to two popular strategies: momentum and value. For momentum strategy we increased the rebalancing frequency to combat volatile market and macro factors. To elaborate, we rebalance the portfolio every week (5 trading days) based on the past month (21 trading days) return ranking. Regarding value strategy, instead of investing in value companies as suggested in Fama-French model, we take long positions in growth companies.

Our primary data resource is Yahoo Finance. We included daily adjusted close price for 20 component companies in S&P 500 (from 2007 to 2022). Within each of the 10 sectors (exclude financials) in the S&P 500, we picked the two with the highest capitalization to stand for the whole sector. Two reasonings here: First, we incorporated 10 sectors to achieve diversification.

Each sector is correlated to certain risk factors, so the big sector pool can diversify relevant risks. Second, as S&P 500 index is weighted by market caps, top two large caps can represent the sector movement best referring to their weights. Notice that we excluded financial sectors from the pool. Compared to other 10 sectors, financials usually hold the lowest P/B Ratio, hence, the value strategy will have a strong preference towards long position in financial sector. In that case, the portfolio cannot be fully diversified and may lack representativeness.

Equity Line is used here to compare strategy performances primarily. other methods such as testing regression and performance statistics are also included. In conclusion, the momentum strategy gave a satisfactory and stable excess return, hence, we recommend investors to follow this strategy to realize effective hedging. The value strategy may need further modification. If multi-strategy needed, investors should choose the Risk Parity method for a maximized hedging result instead of simple equal weighting.

Get Adjusted close price for 20 components

```
library(quantmod)
library(tidyr)
library(dplyr)
stockindex = c('AAPL', 'MSFT',
               'UNH', 'JNJ',
               'AMZN', 'TSLA',
               'GOOGL','META',
               'RTX', 'UPS',
               'PG', 'KO',
               'XOM', 'CVX',
               'NEE', 'DUK',
               'AMT', 'SPG',
               'LIN', 'SHW')
getSymbols(Symbols = stockindex)
```

```
##  [1] "AAPL"  "MSFT"  "UNH"   "JNJ"   "AMZN"  "TSLA"  "GOOGL" "META"  "RTX"
## [10] "UPS"   "PG"    "KO"    "XOM"   "CVX"   "NEE"   "DUK"   "AMT"   "SPG"
## [19] "LIN"   "SHW"
```

```
xx = NULL
for (i in stockindex) {
  allrow = rep(TRUE, nrow(get(i)))
  xx = cbind(xx, subset(get(i),
                        subset = allrow,
                        select = grepl('Adjusted', colnames(get(i)))
  ))
}

xx_Adjusted = xts(xx)
colnames(xx_Adjusted) = stockindex
```

Get Adjusted close price for S&P 500

```
getSymbols(Symbols = 'SPY')
```

```
## [1] "SPY"
```

```
SPY_Adjusted = xts(SPY$SPY.Adjusted)
colnames(SPY_Adjusted) = 'SPY'
```

Get quarterly P/B Ratio for 20 components

```
directory = '/Users/fenglingweng/Desktop/01_JHU/05_Summer/01_AHF/AHF_finalproject/
Components_Statistics'


filetable = lapply(list.files(directory), function(x){
  temp = read.csv(paste0(directory, '/', x))
  temp %>%
    subset(name == 'PbRatio') %>%
    mutate(name = strsplit(x, '_')[[1]][1])
})

xx = data.table::rbindlist(filetable, fill = TRUE)
xx = t(xx)
colnames(xx) = xx[1, ]
xx = xx[c(-1, -2), ]

aa =  as.Date(row.names(xx), tryFormats = 'X%m.%d.%Y')
xx = xts(xx, order.by = aa)

# Next we filter for records later than 2006-12-31 since there are a lot missing v
alues prior to this date.
xx_PbRatio = xx[index(xx) > as.Date('2006-12-31'), ]
```

Compute daily continuously compounded return

```
ret_22 = diff(log(xx_Adjusted)) * 100
ret_SPY = diff(log(SPY_Adjusted)) * 100
```

Momentum Strategy: re-balance every week based on past 1-month (21 trading days) stock return history

```
ret_22_past21 = matrix(NA, nrow = nrow(ret_22), ncol = ncol(ret_22))
ret_22_past21 = xts(ret_22_past21, order.by = index(ret_22))
for (i in 22:nrow(ret_22_past21)) {
  ret_22_past21[i, ] = apply(ret_22[(i-20):i, ], 2, sum)
}
# This step we sum up all past 21 trading days return for each company so we can d
o the sorting next
colnames(ret_22_past21) = stockindex

# Here we define position matrix which assigns -1 or 1 to each stock as long/short
positioning
Mom_position = matrix(0, nrow = nrow(ret_22), ncol = ncol(ret_22))
Mom_position = xts(Mom_position, order.by = index(ret_22))
colnames(Mom_position) = stockindex

# To re-balance the portfolio every week, we sort top 3 high-return companies and
assign positions as 1, and vice versa
for (i in 6:nrow(ret_22_past21)) {
  max3 = order(ret_22_past21[i, ],decreasing=TRUE)[1:3]
  Mom_position[i, max3] = 1
  min3 = order(ret_22_past21[i, ],decreasing=FALSE)[1:3]
  Mom_position[i, min3] = -1
}

# num_interval is the total re-balancing times we can perform
num_interval = floor(nrow(Mom_position)/5)
for (i in 1:(num_interval-1)) {
  for (j in (5*i+1):(5*(i+1))) {
    Mom_position[j, ] = Mom_position[(5*i+1), ]
  }
}
# With the above for loop, we apply the position of the first day in one re-balanc
ing period to the rest 4 days in this period
# To make it straightforward, we copy and paste the first-day position for 4 times
to cover the entire re-balancing period (one week)

Mom_position[(5 * num_interval+1):nrow(Mom_position), ] = Mom_position[(5 * num_in
terval+1), ] # To cover the last several days which may not be a whole week

Mom = data.frame(ret_22 * Mom_position)
Mom$Portfolio = apply(Mom, 1, sum, na.rm = TRUE)/3
```

Value Strategy: rebalance every quarter based on past year average P/B ratio

```
xx_PbRatio = xx_PbRatio[, stockindex]
xx_PbRatio = as.data.frame(lapply(xx_PbRatio, as.numeric))
xx_PbRatio = xts(xx_PbRatio, order.by = index(xx)[index(xx) > as.Date('2006-12-31'
)])

Pbratio_yearave = matrix(NA, nrow = nrow(xx_PbRatio), ncol = ncol(xx_PbRatio))
Pbratio_yearave = xts(Pbratio_yearave, order.by = index(xx_PbRatio))
for (i in 5:nrow(Pbratio_yearave)) {
  Pbratio_yearave[i, ] = apply(xx_PbRatio[(i-3):i, ], 2, mean, na.rm = TRUE)
}

Value_position_quarter = matrix(0, nrow = nrow(xx_PbRatio), ncol = ncol(xx_PbRatio
))
Value_position_quarter = xts(Value_position_quarter, order.by = index(xx_PbRatio))
colnames(Value_position_quarter) = stockindex
for (i in 1:nrow(xx_PbRatio)) {
  max3 = order(Pbratio_yearave[i, ],decreasing=TRUE)[1:3]
  Value_position_quarter[i, max3] = 1
  min3 = order(Pbratio_yearave[i, ],decreasing=FALSE)[1:3]
  Value_position_quarter[i, min3] = -1
}

Value_position = matrix(0, nrow = nrow(ret_22), ncol = ncol(ret_22))
Value_position = xts(Value_position, order.by = index(ret_22))
colnames(Value_position) = stockindex
date_index = c(index(Value_position_quarter), tail(index(ret_22),1))
for (i in 1:(length(date_index)-1)) {
  for (j in as.Date(date_index[i]:date_index[i+1])) {
    temp_index = as.Date(intersect(j, index(Value_position)))
    Value_position[temp_index, ] = Value_position_quarter[date_index[i], ]
  }
}

Value = data.frame(ret_22 * Value_position)
Value$Portfolio = apply(Value, 1, sum, na.rm = TRUE)/3
```

Risk Parity Portfolio (Momentum + Value)

```
EW_Ret = (Mom$Portfolio + Value$Portfolio)/2
Rtns = cbind(Mom$Portfolio, Value$Portfolio, EW_Ret)
colnames(Rtns) = c('Mom', 'Value', 'EW')
cor(Rtns)
```

```
##                Mom       Value        EW
## Mom    1.00000000 -0.02663192 0.7284487
## Value -0.02663192  1.00000000 0.6654574
## EW     0.72844871  0.66545735 1.0000000
```

Use equal-weighted 20 components as the benchmark

```
ew_Rtn22 = apply(ret_22, 1, mean, na.rm = TRUE)/100
cor(ret_SPY[-1], ew_Rtn22[-1])
```

```
##          [,1]
## SPY 0.9574023
```

```
annulizedstd = apply(Rtns, 2, sd) * sqrt(252)

Risk_Mom_weight = 0.682
Risk_Value_weight = 0.318
RP_Rtns = Risk_Mom_weight * Mom$Portfolio + Risk_Value_weight * Value$Portfolio
Rtn_all = cbind(ret_SPY, Mom$Portfolio, Value$Portfolio, RP_Rtns)/100
```

Annulized return

```
n = nrow(Rtn_all)
annulizedrtn = apply(Rtn_all, 2, sum, na.rm = TRUE) * (252/n)
annulizedrtn
```

```
##             SPY   Mom.Portfolio Value.Portfolio         RP_Rtns
##      0.08266285      0.42223207      0.02448440      0.29574831
```

Annulized std

```
annulizedstd = apply(Rtn_all, 2, sd, na.rm = TRUE) * sqrt(252)
annulizedstd
```

```
##             SPY   Mom.Portfolio Value.Portfolio         RP_Rtns
##       0.2050531       0.2935271       0.2694076       0.2156394
```

Sharpe Ratio (Rf = 0)

```
SharpeR <- annulizedrtn/annulizedstd
SharpeR
```

```
##             SPY   Mom.Portfolio Value.Portfolio         RP_Rtns
##      0.40312905      1.43847745      0.09088235      1.37149466
```

Sortino Ratio (MAR = 0)

```
SortinoR <- apply(Rtn_all[-1, ], 2, function(tmp){
  return(mean(tmp)/sqrt(mean(pmin(tmp, 0)^2)))
})
SortinoR
```

```
##             SPY   Mom.Portfolio Value.Portfolio         RP_Rtns
##     0.034930273     0.137889902     0.008015723     0.131977907
```

Omega Ratio (r = 0)

```
OmegaR <- apply(Rtn_all[-1, ], 2, function(tmp){
  return(mean(pmax(tmp, 0))/mean(pmax(-tmp, 0)))
})
OmegaR
```

```
##             SPY   Mom.Portfolio Value.Portfolio         RP_Rtns
##        1.082966        1.304277        1.016483        1.287867
```

Jensen’s alpha

```
Mom.fit = lm(Rtn_all$Mom.Portfolio[-1] ~ ew_Rtn22[-1])
summary(Mom.fit)
```

```
##
## Call:
## lm(formula = Rtn_all$Mom.Portfolio[-1] ~ ew_Rtn22[-1])
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.159963 -0.009158 -0.000155  0.008726  0.153476
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.001794   0.000288   6.230 5.14e-10 ***
## ew_Rtn22[-1] -0.232581   0.023325  -9.971  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01827 on 4029 degrees of freedom
## Multiple R-squared:  0.02408,    Adjusted R-squared:  0.02384
## F-statistic: 99.43 on 1 and 4029 DF,  p-value: < 2.2e-16
```

```
Value.fit = lm(Rtn_all$Value.Portfolio[-1] ~ ew_Rtn22[-1])
summary(Value.fit)
```

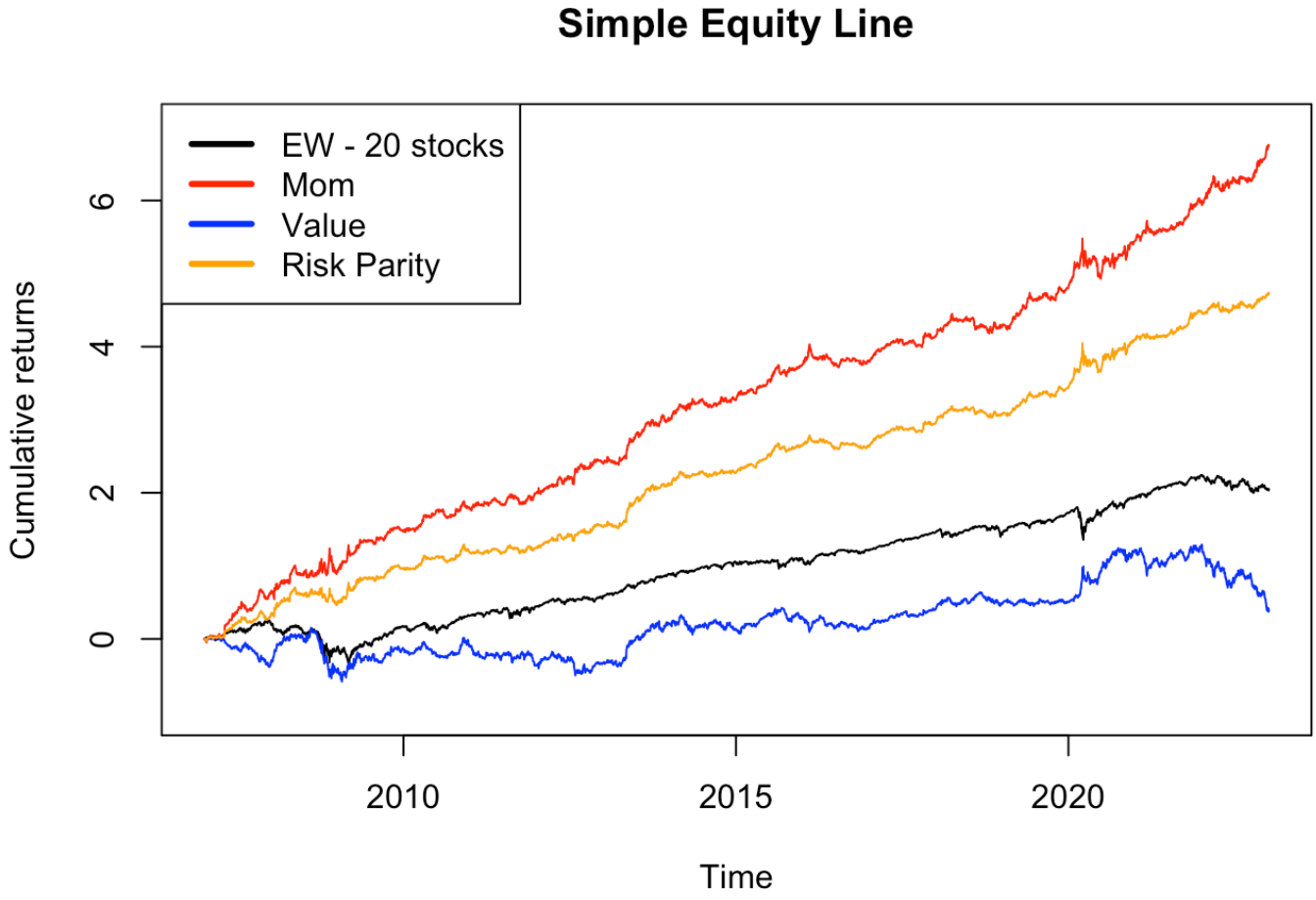
```
##
## Call:
## lm(formula = Rtn_all$Value.Portfolio[-1] ~ ew_Rtn22[-1])
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.144280 -0.008304  0.000327  0.008275  0.160816
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -2.842e-05  2.633e-04  -0.108    0.914
## ew_Rtn22[-1]  2.467e-01  2.132e-02  11.570   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0167 on 4029 degrees of freedom
## Multiple R-squared:  0.03216,    Adjusted R-squared:  0.03192
## F-statistic: 133.9 on 1 and 4029 DF,  p-value: < 2.2e-16
```

```
RP.fit = lm(Rtn_all$RP_Rtns[-1] ~ ew_Rtn22[-1])
summary(RP.fit)
```

```
##
## Call:
## lm(formula = Rtn_all$RP_Rtns[-1] ~ ew_Rtn22[-1])
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.117674 -0.006686 -0.000215  0.006665  0.118208
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.0012147  0.0002136   5.686 1.39e-08 ***
## ew_Rtn22[-1] -0.0801813  0.0172997  -4.635 3.69e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01355 on 4029 degrees of freedom
## Multiple R-squared:  0.005304,   Adjusted R-squared:  0.005057
## F-statistic: 21.48 on 1 and 4029 DF,  p-value: 3.685e-06
```

Equity line

```
plot(index(Rtn_all), cumsum(coalesce(ew_Rtn22, 0)) + ew_Rtn22*0, col = "black", yl
im = c(-1, 7), type = 'l',
     xlab = 'Time', ylab = 'Cumulative returns', main ='Simple Equity Line')
lines(index(Rtn_all)[!is.na(Rtn_all$Mom.Portfolio)], cumsum(Rtn_all$Mom.Portfolio)
, col = "red")
lines(index(Rtn_all)[!is.na(Rtn_all$Value.Portfolio)], cumsum(Rtn_all$Value.Portfo
lio), col = "blue")
lines(index(Rtn_all)[!is.na(Rtn_all$RP_Rtns)], cumsum(Rtn_all$RP_Rtns), col = "ora
nge")
legend('topleft', legend = c('EW - 20 stocks', 'Mom', 'Value', 'Risk Parity'), col
= c('black', 'red', 'blue', 'orange'), lty=1, lwd=3)
```


Simple Equity Line
EW - 20 stocks
Mom
Value
Risk Parity
Cumulative returns
0
2
4
6
2010
2015
2020
Time