COMP226: Slides 12

Risk reward measures

Rahul Savani

rahul.savani@liverpool.ac.uk

Overview

We now study performance measures that combine a measure of **profitability** (reward) with a measure of **risk**, such as the:

- Sharpe Ratio
- Information Ratio
- Sortino Ratio
- Maximum drawdown
- Calmar Ratio

They are **implemented** in the **PerformanceAnalytics** package.

Risk/Reward Performance Measures

When the performance of a trading strategy is measure, we typically aim to **combine different aspects** of performance

In addition to profit/return a measure of **risk** is often incorporated

risk is measured in different ways, e.g.,

- standard deviation of returns (Sharpe ratio)
- downside deviation (Sortino ratio)
- maximum drawdown

Sharpe Ratio

returns	R
risk free rate	R_f
Sharpe Ratio	$\frac{mean(R - R_f)}{\sqrt{variance(R - R_f)}}$

- Very common measure of fund performance
- Ratio of **"reward"** and **"risk"** (captured in this specific and questionable way)

Example

```
library(xts)

dates <- as.yearmon(2000 + seq(0, 11)/12)
dates <- as.Date(dates)

returns <- c(0.01,-0.02,0.05,0.01,-0.01,0.07,0.06,0,0,-0.1,0.1,0.2)
returns <- xts(returns,dates)

Rf <- 0.02
```

```
library(PerformanceAnalytics)

SharpeRatio(returns,Rf,FUN='StdDev')

mean(returns - Rf)/sd(returns-Rf) # for comparison
```

Information Ratio

returns	R
benchmark return	R_b
Information Ratio	$\dfrac{mean(R - R_b)}{\sqrt{variance(R - R_b)}}$

Similar to Sharpe Ratio but allows benchmarks such as **Treasury Bills** or the **S&P 500 index**

Sortino Ratio

returns	R
minimum acceptable return	MAR
Sortino ratio	$\dfrac{mean(R - MAR)}{DownsideDeviation}$

Only penalize deviations below MAR

Downside Deviation =

root mean square of **returns below the MAR**

(see code example that follows for formula)

2 variants of Downside Deviation

1. **subset** method: takes the root mean square only of those deviations where the return is below the MAR

2. **full** method: it includes a 0 in the root mean square for every return above the MAR, and the deviation below the MAR for every other return

Note: because the extra entries in the full method are all zero, the sum in both cases is the same, the only (potential) difference is the divisor in the mean - see examples below

Subset method

```
> DownsideDeviation(returns,Rf,method='subset')
[1] 0.0505682
> returns[returns<Rf]
            [,1]
2000-01-01  0.01
2000-02-01 -0.02
2000-04-01  0.01
2000-05-01 -0.01
2000-08-01  0.00
2000-09-01  0.00
2000-10-01 -0.10
> dreturns <- returns[returns<Rf]
> sqrt(sum((dreturns-Rf)^2)/length(dreturns))
[1] 0.0505682
```

Full method

```
> DownsideDeviation(returns,Rf,method='full')
[1] 0.0386221

> sqrt(sum((dreturns-Rf)^2)/length(returns))
[1] 0.0386221
```

Warning

All three ratios assume that the second moments (i.e. standard deviation) is enough to measure risk.

However it is widely accepted that financial return distributions have "fat tails", making this assumption invalid.

Nonetheless, these ratios are still very commonly used in practice.

Example from Morningstar

Volatility Measures SPY

3-Year | 5-Year | 10-Year | 15-Year

3-Year Trailing	Standard Deviation	Return	Sharpe Ratio	Sortino Ratio	Semi-Standard Deviation
SPY	11.14	13.89	1.12	1.72	—
S&P 500 TR USD	11.18	14.02	1.13	1.73	—
Category: LB	11.54	12.50	0.98	1.51	—

01/31/2019

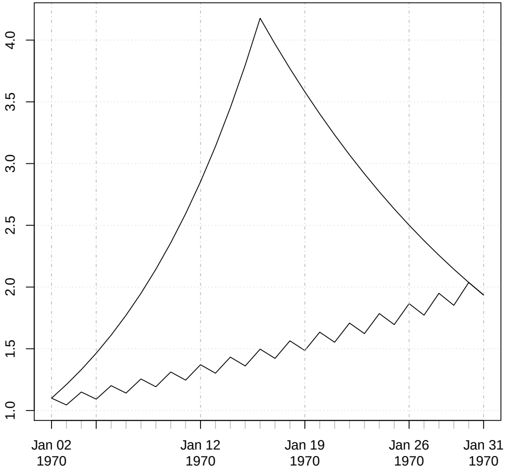
http://performance.morningstar.com/funds/etf/ratings-risk.action?t=SPY

Issues with standard deviation

As a measure of risk it suffers from several drawbacks:

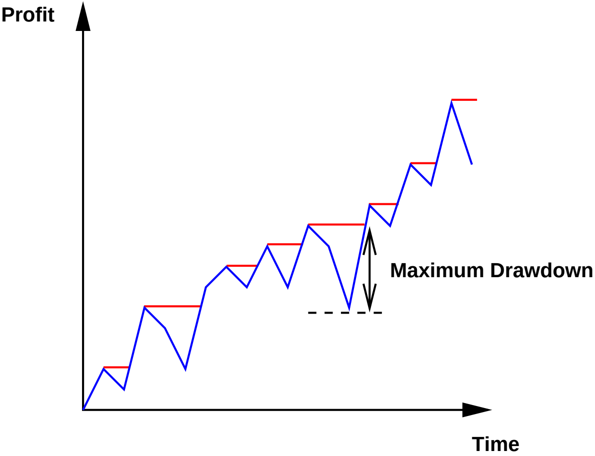
- Symmetric with respect to upside and downside
  - Upward and downward changes contribute equally
  - Investors typically associate risk only with the downside
  - Hence sometimes the Sortino ratio is used
- Oblivious to **sequence** of gains and losses
  - Does not recognize strings of losses (i.e. **drawdowns**)

Two equity curves with the same Sharpe Ratios



Maximum drawdown

decline from historical peak



Drawdown: profit/loss

Suppose the cumulative profit/loss at time t is

$$S_t \quad t = 1, \ldots, n$$

The **running maximum cumulative proft/loss** is

$$M_t = \max_{u \in [1,t]} S_u$$

The **drawdown** is

$$D_t = M_t - S_t$$

The **maximum drawdown** at period t is

$$MDD_t = \max_{u \in [1,t]} D_u$$

The **maximum drawdown** for the whole period 1 to n is

$$MDD_n$$

Wealth index vs. Cumulative simple returns

Suppose the simple returns for periods 1 to n are

$$R_t \quad t = 1, \ldots, n$$

The **cumulative return** in period t is

$$\left(\prod_{i = 1, \ldots, t} (1 + R_i) \right) - 1$$

The **wealth index** is the same **except we do not subtract 1**:

$$W_t = \prod_{i = 1, \ldots, t} (1 + R_i)$$

The drawdown calculation uses the wealth index

Drawdown: returns

The **running maximum of the wealth index** is

$$M_t = \max_{u \in [1,t]} W_t$$

The **drawdown** is then

$$D_t = -(\frac{W_t}{M_t} - 1) = 1 - \frac{W_t}{M_t}$$

The reason for the minus sign is to make the drawdown positive, so that it is the same sign as in the profit/loss case defined earlier (and the term **maximum** is accurate)

The **maximum drawdown** in period t is (same as for profit case)

$$MDD_t = \max_{u \in [1,t]} D_u$$

This returns version is implemented in `PerformanceAnalytics`

Example with profits

```
> source('drawdowns_pnl.R',echo=T)

> simple_diffs <- c(10,5,20,-7,-20,6,-30,20,30,-40)
> cumulative_pnl <- cumsum(simple_diffs)
> running_max <- cummax(cumulative_pnl)
> drawdowns  <- running_max - cumulative_pnl

> print(rbind(running_max,cumulative_pnl,drawdowns))
               [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
running_max      10   15   35   35   35   35   35   35   35    35
cumulative_pnl   10   15   35   28    8   14  -16    4   34    -6
drawdowns         0    0    0    7   27   21   51   31    1    41

> cat('The maxdrawdown is',max(drawdowns),'\n')
The maxdrawdown is 51
```

Example with returns

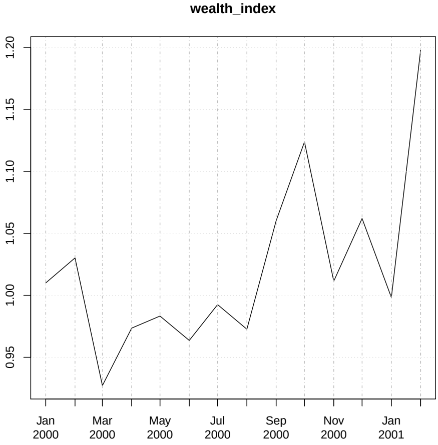
```r
library(xts)
library(PerformanceAnalytics)

returns <- c(0.01,0.02,-0.1,0.05,0.01,-0.02,0.03,
                    -0.02,0.09,0.06,-0.1,0.05,-0.06,0.2)

dates <- as.yearmon(2000 + seq(0, (length(returns)-1))/12)
dates <- as.Date(dates)

returns <- xts(returns,dates)
```

Wealth index



Drawdowns

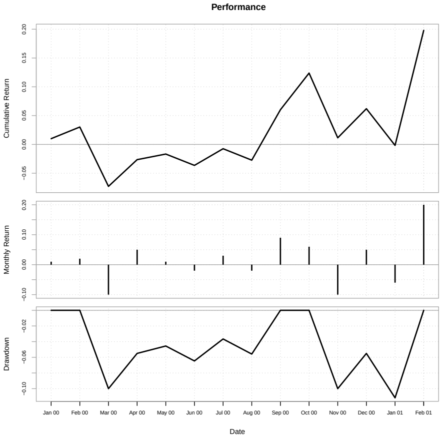
```
> wealth_index <- cumprod(1+returns)
> drawdowns <- 1 - wealth_index/cummax(wealth_index)
> print(tail(drawdowns))
             [,1]
2000-09-01 0.0000
2000-10-01 0.0000
2000-11-01 0.1000
2000-12-01 0.0550
2001-01-01 0.1117
2001-02-01 0.0000

> cat('The maxdrawdown is',max(drawdowns),'\n')
The maxdrawdown is 0.1117
```

```
> table.Drawdowns(returns) # from PerformanceAnalytics
        From     Trough         To   Depth Length To Trough Recovery
1 2000-11-01 2001-01-01 2001-02-01 -0.1117      4         3        1
2 2000-03-01 2000-03-01 2000-09-01 -0.1000      7         1        6
Warning message:
In table.Drawdowns(returns, top = 10) : Only 2 available in the data.

> maxDrawdown(returns) # from PerformanceAnalytics
[1] 0.1117
```

```
> charts.PerformanceSummary(returns)
```



Calmar Ratio

$$\frac{FinalCumulativeReturn}{MaximumDrawdown}$$

```
> PerformanceAnalytics::CalmarRatio(returns)
                 [,1]
Calmar Ratio 1.498954
```

Exercise

empirically compare the Calmar Ratio with the Sharpe Ratio

Summary

We have studied **simple and log returns** as **building blocks of performance measurement**

Then we have studied a number of performance measures that can be used to measure the **relative performance** of trading strategies

All of them are a **ratio** of some **measure of profitability** to some **measure of risk**

These performance measures can be used as **fitness functions** for optimization, which we will discuss later.