irr-i-ardly-kner-er

May 27, 2023

1 IRR? I 'Ardly Kner 'Er!

@sparshsah

1.1 Quants are quants, and IRR's are IRR's, and ne'er the twain shall meet

You are a quant running a global small-cap single-name stock-selection strategy. As a quant, you run an optimizer. Let's ignore risk and focus on expected returns. (By "ignore risk", I mean pretend that every passive asset has equal risk, whether you want to define that as volatility, or value-at-risk, or expected shortfall, or whatever.)

You rebalance once every two weeks. (Exactly two weeks happens to be the schedule of a certain unnamed hedge-fund manager in Greenwich, but the idea is that if you rebalanced every day you'd incur so much tcost as to swamp your gross returns. Unless you're Bernie Madoff. Or Medallion.)

1.1.1 The setup

You have two assets, such that (measured over the next two weeks, but quoted in annualized terms, and using log returns because they're more convenient to reason formally about): * Asset A has 20% ex-ante IRR, * Asset B has 20% ex-ante IRR, * and cash earns 5% guaranteed.

More specifically, assuming that both A and B are selling for $1/share right now (see Appendix to confirm that both these IRR's are, indeed, exactly 20%): * If you buy 1 share of Asset A, you expect to sell it for about $1.0077 (see Appendix for exact price) at the end of the second week; whereas, * If you buy 1 share of Asset B, you expect to collect a dividend of 50.00c at the end of the first week, then sell the ex-dividend share for about 50.58c (see Appendix for exact price) at the end of the second week.

1.1.2 The result

Of course, to run an optimizer, you have to summarize the assets somehow. So, you choose to tell the optimize their IRR's – And since they're identical, the optimizer gets to choose at random. Suppose you start with $1 and face the same opportunity set fortnight after fortnight for a year.

- If your optimizer kept randomly choosing Asset A, you'd end the year with $1.20.
- If your optimizer kept randomly choosing Asset B, you'd end the year with… just $1.16!

So a quant can't use IRR's (20% and 20% in this case) to optimize – She really must use CAGR's (20% and 16% in this case).

1.2 Anything you can do I can do better

The standard quibble with CAGR's is that (a) you have to specify some terminal timestep, and (b) you have to specify a cash rate. Both these inputs can seem arbitrary or, at the very least, hard to pin down. IRR's require no such inputs. This is true. And in the case above, because the timescales were so small, it feels (to me) hard to tell why the two are so at-odds with each other. So there's some tension: I'm asking you to use this inelegant thing, that I can't really even satisfactorily explain why it gives such different results.

1.2.1 No you can't

In fact, I've heard people argue that, at longer (private-equity-like) time scales, taking into account WAL (weighted-average life) sets you straight. (Never mind that they can never actually articulate *how*, formally, I'm meant to "take into account WAL" – Do I divide by it? Add it? Take it to the zoo?) But how about this scenario, with two bonds, each with face value $1, an eye-popping IRR of 50%, and WAL 10Y: * Bond A pays 50c annual interest at the end of each of the next ten years, and repays its $1 of principal along with the last interest payment at the end of the tenth year, * Bond B pays a one-time $1.47 interest payment at the end of the first year, then nothing until it repays its principal at the end of the tenth year.

The goal here isn't to be realistic, it's to gain some intuition. * With Bond A, if you do nothing with the cash inflows (or, if you prefer, assume the risk-free rate is zero), you end with $6 of wealth. * With Bond B, under the same assumption, you end with… $2.47.

1.2.2 Yes I can

Of course, with Bond B you get an extra 97c at the end of the first year, and you can "do something useful with it". Indeed you might even prefer Bond B for this reason*. My point isn't that Bond B is a bad investment…

1.2.3 No you can't, no you *can't*!

… My point is that you can't tell a (mean-variance) optimizer "Both bonds have IRR 50%, but Bond B gets a bigger lumpsum early-on so you can do something useful with that, whereas Bond A keeps earning interest for the full ten years". When you're trying to optimize over the global universe of 1,000's of stocks, you need to decide how frequently you will rebalance, and tell the optimizer your predicted CAGR's over that period. And the "cash rate" is whatever JPMorganChase will pay on the dividends you collect along the way. So in practical applications, neither the timeframe nor the cash rate are arbitrary.

* *Me personally I'd suggest that you could still take Bond A, collect your 50c of interest, and sell the remaining 9-year bond to raise an extra 100c of cash… In fact, because shorter-term bonds tend to carry lower annualized yields than longer-term bonds, you could probably raise even a bit more than face value when you sell your bond, say 102c. That is, if other 9-year bonds carry 48% prevailing yields, then rational buyers will be happy to pay a premium for your 9-year bond which carries a 50% yield. But I'm also sympathetic to the fact that, as hard as it is to predict future cashflows, it's even harder to predict how the animal spirits will price those cashflows.*

1.3 Appendix: Calculations

[31]:

```
from typing import Final

import math

import numpy as np

# annualized, logarithmic
IRR: Final[float] = 0.20
CASH: Final[float] = 0.05
PERIODS_PER_YEAR: Final[int] = 52


def get_npv(cfs: list[float]) -> float:
    return sum(
        cf * math.exp(-IRR* t/PERIODS_PER_YEAR)
        for (t, cf) in enumerate(cfs)
    )


def confirm_irr(cfs: list[float]) -> None:
    npv = get_npv(cfs=cfs)
    if not np.isclose(npv, 0):
        raise ValueError(npv)
    cfs_fmt = ", ".join(f"${cf:.4f}" for cf in cfs)
    print(f"[ {cfs_fmt} ] has {IRR:.2%} annualized logarithmic IRR.")
```

1.3.1 Asset A

Confirm it has 20% IRR:

[27]:

```
cfs_a = [
    -1,
    0,
    math.exp(IRR * 2/PERIODS_PER_YEAR),
]

confirm_irr(cfs=cfs_a)
```

```
[ $-1.0000, $0.0000, $1.0077 ] has 20.00% annualized logarithmic IRR.
```

Value at next rebalance:

[60]:

```
value_a_terminal = cfs_a[2]

f"${value_a_terminal:.4f}"
```

[60]:

```
'$1.0077'
```

Implied CAGR:

[39]:
```
cagr_a = math.log(value_a_terminal) * PERIODS_PER_YEAR/2

f"{cagr_a:.2%}"
```

[39]:
```
'20.00%'
```

1.3.2 Asset B

Confirm it has 20% IRR:

[56]:
```
_dividend_b = 0.50

cfs_b = [
    -1,
    _dividend_b,
    # solve for the necessary PX_LAST
    (
        (
            1 - _dividend_b * math.exp(-IRR * 1/PERIODS_PER_YEAR)
        ) / (
            math.exp(-IRR * 2/PERIODS_PER_YEAR)
        )
    ),
]

confirm_irr(cfs=cfs_b)
```

```
[ $-1.0000, $0.5000, $0.5058 ] has 20.00% annualized logarithmic IRR.
```

Solve for value at next rebalance, assuming dividend earns cash rate:

[59]:
```
value_b_terminal = (
    cfs_b[1] * math.exp(CASH * 1/PERIODS_PER_YEAR)
    +
    cfs_b[2]
)

f"${value_b_terminal:.4f}"
```

[59]:
```
'$1.0063'
```

Implied CAGR:

[58]:
```
cagr_b = math.log(value_b_terminal) * PERIODS_PER_YEAR/2

f"{cagr_b:.2%}"
```

[58]:
```
'16.27%'
```