binary-sharpe

April 8, 2022

AUTHOR: SPARSHSAH

1 SHARPE OF A BINARY SIGNAL

1.1 Setting

We have a single asset with return $r_t \in \{-1\%, +1\%\}$, with each possibility equally probable. In fact, for convenience (it won't matter in our setting), let's normalize this to $r_t \in \{-1, +1\}$. (If it helps you, pretend we have levered the asset 10x.) We also have a signal $s_t \in \{-1, +1\}$. The signal is "right" with probability p.

Unconditionally,

$$\Pr[r_t = -1] = \Pr[r_t = +1] = 0.50 = \Pr[s_t = -1] = \Pr[s_t = +1].$$

Jointly,

$$\Pr[\operatorname{sgn}(r_t) = \operatorname{sgn}(s_t)] = p,$$

that is, conditionally,

$$\Pr[r_t = -1 \mid s_t = -1] = \Pr[s_t = -1 \mid r_t = -1] =: p := \Pr[r_t = +1 \mid s_t = +1] = \Pr[s_t = +1 \mid r_t = +1]$$

and

$$\Pr[r_t = -1 \mid s_t = +1] = \Pr[s_t = +1 \mid r_t = -1] =: 1-p := \Pr[r_t = +1 \mid s_t = -1] = \Pr[s_t = -1 \mid r_t = +1].$$

1.2 Analysis

I want to know this signal's ex-ante Sharpe. First, let's calculate the ex-ante correlation between the asset's return and this signal, noting that $\mu_r = 0 = \mu_s$ and $\sigma_r = 1 = \sigma_s$, so that

$$\rho := \operatorname{Corr}(r_t, s_t) = \operatorname{Cov}(r_t, s_t) = \mathbb{E}[r_t s_t].$$

We'll find it handy in a moment to know (by Adam's Law with extra conditioning)

$$\mathbb{E}[r_t s_t \mid s_t = -1]$$

$$= \mathbb{E}[r_t s_t \mid r_t = -1, s_t = -1] \Pr[r_t = -1 \mid s_t = -1] + \mathbb{E}[r_t s_t \mid r_t = +1, s_t = -1] \Pr[r_t = +1 \mid s_t = -1]$$

$$= (-1)(-1)p + (+1)(-1)(1-p)$$

$$= p + (-1)(1-p) = p - 1 + p = \boxed{2p-1},$$

and symmetrically for $\mathbb{E}[r_t s_t \mid s_t = +1]$.

Now we have again by Adam's Law

$$\begin{aligned}
&\mathbb{E}[r_t s_t] \\
&= \mathbb{E}[r_t s_t \mid s_t = -1] \Pr[s_t = -1] + \mathbb{E}[r_t s_t \mid s = +1] \Pr[s_t = +1] \\
&= (2p-1)0.50 + (2p-1)0.50 = \boxed{2p-1}.
\end{aligned}$$

So, ρ is linear in p:

$$\begin{aligned}
(\rho \mid p = 0.00) &= -1.00 \\
(\rho \mid p = 0.25) &= -0.50 \\
(\rho \mid p = 0.50) &= 0.00 \\
(\rho \mid p = 0.75) &= +0.50 \\
(\rho \mid p = 1.00) &= +1.00.
\end{aligned}$$

So, by my result here, your signal needs to be right only 53.1% of the time to yield 1.00 annualized Sharpe!

[1]:
```python
from typing import Tuple, Optional
import pandas as pd
import numpy as np

P: float = 0.531
BIT_CHOICES: Tuple[int] = (-1, +1)
DAYS_PER_YEAR: int = 261
T: int = 10_000_000


def _calc_rho(p: float=P) -> float:
    rho = 2*p - 1
    return rho

def _calc_expected_sharpe(rho: float, daycount: float=DAYS_PER_YEAR):
    """https://github.com/sparshsah/foggy-demo/blob/main/demo/finance/
↪sharpe-from-correl.pdf"""
    return rho * DAYS_PER_YEAR**0.5

def calc_expected_sharpe(p: float=P) -> float:
    rho = _calc_rho(p=p)
    expected_sharpe = _calc_expected_sharpe(rho=rho)
    return expected_sharpe


def get_pnl(sigs: pd.Series, rets: pd.Series) -> pd.Series:
    pnl = sigs * rets
    return pnl
```

```python
def calc_sharpe(pnl: np.array, daycount: float=DAYS_PER_YEAR) -> float:
    er = np.mean(pnl)
    vol = np.std(pnl)
    sr = er / vol
    ann_sr = sr * daycount**0.5
    return ann_sr


def _get_bit(given: int=None) -> int:
    if given is None:
        return np.random.choice(BIT_CHOICES)
    else:
        b = np.random.rand()
        return given if b < P else -given

def _gen_sig() -> int:
    return _get_bit()

def _gen_ret(sig: int) -> int:
    return _get_bit(given=sig)


def gen_sigs() -> pd.Series:
    sigs = np.random.choice(BIT_CHOICES, size=T)
    sigs = pd.Series(sigs)
    return sigs

def gen_rets(sigs: pd.Series) -> pd.Series:
    rets = sigs.apply(_gen_ret)
    return rets

def gen_pnl() -> pd.Series:
    sigs = gen_sigs()
    rets = gen_rets(sigs=sigs)
    pnl = get_pnl(sigs=sigs, rets=rets)
    return pnl


def sim() -> float:
    pnl = gen_pnl()
    sharpe = calc_sharpe(pnl=pnl)
    return sharpe


def main():
    np.random.seed(42)
    expected_sharpe = calc_expected_sharpe()
```

```python
    sim_sharpe = sim()
    msg = f"Expected Sharpe: {expected_sharpe: .2f}\nSimulated Sharpe:␣
↪{sim_sharpe: .2f}"
    print(msg)

main()
```

```
Expected Sharpe:  1.00
Simulated Sharpe:  1.00
```