Using SharpeR

Steven E. Pav *

December 18, 2024

Abstract

The SharpeR package provides basic functionality for testing significance of the Sharpe ratio of a series of returns, and of the Markowitz portfolio on a number of possibly correlated assets.[16] The goal of the package is to make it simple to estimate profitability (in terms of risk-adjusted returns) of strategies or asset streams.

1 The Sharpe ratio and Optimal Sharpe ratio

Sharpe defined the 'reward to variability ratio', now known as the 'Sharpe ratio', as the sample statistic

$$\hat{\zeta} = \frac{\hat{\mu}}{\hat{\sigma}},$$

where $\hat{\mu}$ is the sample mean, and $\hat{\sigma}$ is the sample standard deviation. [16] The Sharpe ratio was later redefined to include a 'risk-free' or 'disastrous rate of return': $\hat{\zeta} = (\hat{\mu} - r_0)/\hat{\sigma}$.

It is little appreciated in quantitative finance that the Sharpe ratio is identical to the sample statistic proposed by Gosset in 1908 to test for zero mean when the variance is unknown. [5] The 't-test' we know today, which includes an adjustment for sample size, was formulated later by Fisher. [4] Knowing that the Sharpe ratio is related to the t-statistic provides a 'natural arbitrage,' since the latter has been extensively studied. Many of the interesting properties of the t-statistic can be translated to properties about the Sharpe ratio.

Also little appreciated is that the multivariate analogue of the t-statistic, Hotelling's T^2, is related to the Markowitz portfolio. Consider the following portfolio optimization problem:

$$\max_{\hat{\boldsymbol{\nu}}:\hat{\boldsymbol{\nu}}^\top \hat{\boldsymbol{\Sigma}} \hat{\boldsymbol{\nu}} \leq R^2} \frac{\hat{\boldsymbol{\nu}}^\top \hat{\boldsymbol{\mu}} - r_0}{\sqrt{\hat{\boldsymbol{\nu}}^\top \hat{\boldsymbol{\Sigma}} \hat{\boldsymbol{\nu}}}}, \tag{1}$$

where $\hat{\boldsymbol{\mu}}$, $\hat{\boldsymbol{\Sigma}}$ are the sample mean vector and covariance matrix, r_0 is the risk-free rate, and R is a cap on portfolio 'risk' as estimated by $\hat{\boldsymbol{\Sigma}}$. (Note this differs from the traditional definition of the problem which imposes a 'self-financing constraint' which does not actually bound portfolio weights.) The solution to this problem is

$$\hat{\boldsymbol{\nu}}_* =_{\mathrm{df}} \frac{R}{\sqrt{\hat{\boldsymbol{\mu}}^\top \hat{\boldsymbol{\Sigma}}^{-1} \hat{\boldsymbol{\mu}}}} \hat{\boldsymbol{\Sigma}}^{-1} \hat{\boldsymbol{\mu}}.$$

---

*shabbychef@gmail.com

The Sharpe ratio of this portfolio is

$$\hat{\zeta}_* =_{\mathrm{df}} \frac{\hat{\boldsymbol{\nu}}_*^\top \hat{\boldsymbol{\mu}} - r_0}{\sqrt{\hat{\boldsymbol{\nu}}_*^\top \hat{\boldsymbol{\Sigma}} \hat{\boldsymbol{\nu}}_*}} = \sqrt{\hat{\boldsymbol{\mu}}^\top \hat{\boldsymbol{\Sigma}}^{-1} \hat{\boldsymbol{\mu}}} - \frac{r_0}{R} = \sqrt{T^2/n} - \frac{r_0}{R}, \tag{2}$$

where T^2 is Hotelling's statistic, and n is the number of independent observations (e.g., 'days') used to construct $\hat{\boldsymbol{\mu}}$. The term r_0/R is a deterministic 'drag' term that merely shifts the location of $\hat{\zeta}_*$, and so we can (mostly) ignore it when testing significance of $\hat{\zeta}_*$.

Under the (typically indefensible) assumptions that the returns are generated *i.i.d.* from a normal distribution (multivariate normal in the case of the portfolio problem), the distributions of $\hat{\zeta}$ and $\hat{\zeta}_*^2$ are known, and depend on the sample size and the population analogues, ζ and ζ_*^2. In particular, they are distributed as rescaled non-central t and F distributions. Under these assumptions on the generating processes, we can perform inference on the population analogues using the sample statistics.

The importance of each of these assumptions (*viz.* homoskedasticity, independence, normality, *etc.*) can and should be checked. [12, 14] The reader must be warned that this package is distributed without any warranty of any kind, and in no way should any analysis performed with this package be interpreted as implicit investment advice by the author(s).

The units of $\hat{\mu}$ are 'returns per time,' while those of $\hat{\sigma}$ are 'returns per square root time.' Consequently, the units of $\hat{\zeta}$ are 'per square root time.' Typically the Sharpe ratio is quoted in 'annualized' terms, *i.e.*, $\mathrm{yr}^{-1/2}$, but the units are omitted. I believe that units should be included as it avoids ambiguity, and simplifies conversions.

There is no clear standard whether arithmetic or geometric returns should be used in the computation of the Sharpe ratio. Since arithmetic returns are always greater than the equivalent geometric returns, one would suspect that arithmetic returns are *always* used when advertising products. However, I suspect that geometric returns are more frequently used in the analysis of strategies. Geometric returns have the attractive property of being 'additive', meaning that the geometric return of a period is the sum of those of subperiods, and thus the sign of the arithmetic mean of some geometric returns indicates whether the final value of a strategy is greater than the initial value. Oddly, the arithmetic mean of arithmetic returns does not share this property.

On the other hand, arithmetic returns are indeed additive *contemporaneously*: if $\boldsymbol{x}$ is the vector of arithmetic returns of several stocks, and $\hat{\boldsymbol{\nu}}$ is the dollar proportional allocation into those stocks at the start of the period, then $\boldsymbol{x}^\top \hat{\boldsymbol{\nu}}$ is the arithmetic return of the portfolio over that period. This holds even when the portfolio holds some stocks 'short.' Often this portfolio accounting is misapplied to geometric returns without even an appeal to Taylor's theorem.

For more details on the Sharpe ratio and portfolio optimization, see the vignette, "Notes on the Sharpe ratio" distributed with this package.

2 Using the sr Class

An `sr` object encapsulates one or more Sharpe ratio statistics, along with the degrees of freedom, the rescaling to a t statistic, and the annualization and units

information. One can simply stuff this information into an `sr` object, but it is more straightforward to allow `as.sr` to compute the Sharpe ratio for you.

```
library(SharpeR)
# suppose you computed the Sharpe of your
# strategy to be 1.3 / sqrt(yr), based on 1200
# daily observations.  store them as follows:
my.sr <- sr(sr = 1.3, df = 1200 - 1, ope = 252, epoch = "yr")
print(my.sr)

##        SR/sqrt(yr) Std. Error t value Pr(>t)
## Sharpe        1.30       0.46     2.8 0.0023 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# multiple strategies can be tracked as well.
# one can attach names to them.
srstats <- c(0.5, 1.2, 0.6)
dim(srstats) <- c(3, 1)
rownames(srstats) <- c("strat. A", "strat. B", "benchmark")
my.sr <- sr(srstats, df = 1200 - 1, ope = 252, epoch = "yr")
print(my.sr)

##           SR/sqrt(yr) Std. Error t value Pr(>t)
## strat. A         0.50       0.46     1.1 0.1377
## strat. B         1.20       0.46     2.6 0.0045 **
## benchmark        0.60       0.46     1.3 0.0953 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Throughout, `ope` stands for 'Observations Per Epoch', and is the (average) number of returns observed per the annualization period, called the `epoch`. At the moment there is not much hand holding regarding these parameters: no checking is performed for sane values.

The `as.sr` method will compute the Sharpe ratio for you, from numeric, `data.frame`, `xts` or `lm` objects. In the latter case, it is assumed one is performing an attribution model, and the statistic of interest is the fit of the (`Intercept`) term divided by the residual standard deviation. Here are some examples:

```
set.seed(as.integer(charToRaw("set the seed")))
# Sharpe's 'model': just given a bunch of
# returns.
returns <- rnorm(253 * 8, mean = 3e-04, sd = 0.01)
asr <- as.sr(returns, ope = 253, epoch = "yr")
print(asr)

##   SR/sqrt(yr) Std. Error t value Pr(>t)
## x        0.56       0.35     1.6  0.056 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# a data.frame with a single strategy
asr <- as.sr(data.frame(my.strategy = returns), ope = 253,
    epoch = "yr")
print(asr)
```

```
##             SR/sqrt(yr) Std. Error t value Pr(>t)
## my.strategy        0.56       0.35     1.6  0.056 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

When a `data.frame` with multiple columns is given, the Sharpe ratio of each is computed, and they are all stored:

```
# a data.frame with multiple strategies
asr <- as.sr(data.frame(strat1 = rnorm(253 * 8), strat2 = rnorm(253 *
    8, mean = 4e-04, sd = 0.01)), ope = 253, epoch = "yr")
print(asr)

##        SR/sqrt(yr) Std. Error t value Pr(>t)
## strat1      -0.043      0.354   -0.12  0.549
## strat2       0.573      0.354    1.62  0.053 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Here is an example using `xts` objects. In this case, if the `ope` is not given, it is inferred from the time marks of the input object.

```
require(quantmod)
# get price data, compute log returns on adjusted
# closes
get.ret <- function(sym, warnings = FALSE, ...) {
    # getSymbols.yahoo will barf sometimes; do a
    # trycatch
    trynum <- 0
    while (!exists("OHCLV") && (trynum < 7)) {
        trynum <- trynum + 1
        try(OHLCV <- getSymbols(sym, auto.assign = FALSE,
            warnings = warnings, ...), silent = TRUE)
    }
    adj.names <- paste(c(sym, "Adjusted"), collapse = ".",
        sep = "")
    if (adj.names %in% colnames(OHLCV)) {
        adj.close <- OHLCV[, adj.names]
    } else {
        # for DJIA from FRED, say.
        adj.close <- OHLCV[, sym]
    }
    rm(OHLCV)
    # rename it
    colnames(adj.close) <- c(sym)
    adj.close <- adj.close[!is.na(adj.close)]
    lrets <- diff(log(adj.close))
    # chop first
    lrets[-1, ]
}
get.rets <- function(syms, ...) {
    some.rets <- do.call("cbind", lapply(syms, get.ret,
        ...))
}
```

```
require(quantmod)
# quantmod::periodReturn does not deal properly
# with multiple columns, and the straightforward
# apply(mtms,2,periodReturn) barfs
my.periodReturn <- function(mtms, ...) {
    per.rets <- do.call(cbind, lapply(mtms, function(x) {
        retv <- periodReturn(x, ...)
        colnames(retv) <- colnames(x)
        return(retv)
    }))
}
# convert log return to mtm, ignoring NA
lr2mtm <- function(x, ...) {
    x[is.na(x)] = 0
    exp(cumsum(x))
}
```

```
some.rets <- get.rets(c("IBM", "AAPL", "XOM"), from = "2007-01-01",
    to = "2013-01-01")
print(as.sr(some.rets))

##      SR/sqrt(yr) Std. Error t value Pr(>t)
## IBM         0.54       0.41    1.31  0.095 .
## XOM         0.15       0.41    0.36  0.360
## AAPL        0.83       0.41    2.02  0.022 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The annualization of an `sr` object can be changed with the `reannualize` method. The name of the epoch and the observation rate can both be changed. Changing the annualization will not change statistical significance, it merely changes the units.

```
yearly <- as.sr(some.rets[, "XOM"])
monthly <- reannualize(yearly, new.ope = 21, new.epoch = "mo.")
print(yearly)

##      SR/sqrt(yr) Std. Error t value Pr(>t)
## XOM         0.15       0.41    0.36   0.36

# significance should be the same, but units
# changed.
print(monthly)

##      SR/sqrt(mo.) Std. Error t value Pr(>t)
## XOM        0.042      0.118    0.36   0.36
```

2.1   Attribution Models

When an object of class `lm` is given to `as.sr`, the fit `(Intercept)` term is divided by the residual volatility to compute something like the Sharpe ratio. In

terms of Null Hypothesis Significance Testing, nothing is gained by summarizing the `sr` object instead of the `lm` object. However, confidence intervals on the Sharpe ratio are quoted in the more natural units of reward to variability, and in annualized terms (or whatever the epoch is.)

As an example, here I perform a CAPM attribution to the monthly returns of `AAPL`. Note that the statistical significance here is certainly tainted by selection bias, a topic beyond the scope of this note.

```r
# get the returns (see above for the function)
aapl.rets <- get.rets(c("AAPL", "SPY"), from = "2003-01-01",
    to = "2013-01-01")
# make them monthly:
mo.rets <- my.periodReturn(lr2mtm(aapl.rets), period = "monthly",
    type = "arithmetic")
rm(aapl.rets)  # cleanup
# look at both of them together:
both.sr <- as.sr(mo.rets)
print(both.sr)

##      SR/sqrt(yr) Std. Error t value  Pr(>t)
## SPY         0.51       0.32     1.6   0.055 .
## AAPL        1.36       0.33     4.3 1.7e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# confindence intervals on the Sharpe:
print(confint(both.sr))

##      2.5 % 97.5 %
## SPY  -0.12    1.1
## AAPL  0.71    2.0

# perform a CAPM attribution, using SPY as 'the
# market'
linmod <- lm(AAPL ~ SPY, data = mo.rets)
# convert attribution model to Sharpe
CAPM.sr <- as.sr(linmod, ope = both.sr$ope, epoch = "yr")
# statistical significance does not change
# (though note the sr summary prints a 1-sided
# p-value)
print(summary(linmod))

##
## Call:
## lm(formula = AAPL ~ SPY, data = mo.rets)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.27094 -0.06403  0.00429  0.05051  0.30164
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.03370    0.00841    4.01  0.00011 ***
## SPY          1.31329    0.19469    6.75    6e-10 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.091 on 118 degrees of freedom
## Multiple R-squared:  0.278,Adjusted R-squared:  0.272
## F-statistic: 45.5 on 1 and 118 DF,  p-value: 5.98e-10

print(CAPM.sr)

##         SR/sqrt(yr) Std. Error t value  Pr(>t)
## linmod         1.28       0.33       4 5.4e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# the confidence intervals tell the same story,
# but in different units:
print(confint(linmod, "(Intercept)"))

##             2.5 % 97.5 %
## (Intercept) 0.017   0.05

print(confint(CAPM.sr))

##        2.5 % 97.5 %
## linmod  0.63    1.9
```

2.2 Testing Sharpe and Power

The function **sr_test** performs one- and two-sample tests for significance of Sharpe ratio. Paired tests for equality of Sharpe ratio can be performed via the **sr_equality_test**, which applies the tests of Leung *et al.* or of Wright *et al.* [10, 19]

```
# get the sector 'spiders'
secto.rets <- get.rets(c("XLY", "XLE", "XLP", "XLF",
    "XLV", "XLI", "XLB", "XLK", "XLU"), from = "2003-01-01",
    to = "2013-01-01")
# make them monthly:
mo.rets <- my.periodReturn(lr2mtm(secto.rets), period = "monthly",
    type = "arithmetic")
# one-sample test on utilities:
XLU.monthly <- mo.rets[, "XLU"]
print(sr_test(XLU.monthly), alternative = "two.sided")

##
##  One Sample sr test, exact method
##
## data:  XLU.monthly
## t = 2, df = 119, p-value = 0.02
## alternative hypothesis: true signal-noise ratio is not equal to 0
## sample estimates:
##     [,1]
## XLU 0.22
```

```
## attr(,"names")
## [1] "Sharpe ratio of XLU.monthly"

# test for equality of Sharpe among the different
# spiders
print(sr_equality_test(as.matrix(secto.rets)))

##
##  test for equality of Sharpe ratio, via chisq test
##
## data:  as.matrix(secto.rets)
## T2 = 7, contrasts = 8, p-value = 0.6
## alternative hypothesis: true sum squared contrasts of SNR is not equal to 0

# perform a paired two-sample test via sr_test:
XLF.monthly <- mo.rets[, "XLF"]
print(sr_test(x = XLU.monthly, y = XLF.monthly, ope = 12,
    paired = TRUE))

##
##  Paired sr-test
##
## data:  XLU.monthly and XLF.monthly
## t = 2, df = 119, p-value = 0.07
## alternative hypothesis: true difference in signal-noise ratios is not equal to 0
## sample estimates:
## difference in Sharpe ratios
##                        0.68
```

3 Using the `sropt` Class

The class `sropt` stores the 'optimal' Sharpe ratio, which is that of the optimal ('Markowitz') portfolio, as defined in Equation 1, as well as the relevant degrees of freedom, and the annualization parameters. Again, the constructor can be used directly, but the helper function is preferred:

```
set.seed(as.integer(charToRaw("7bf4b86a-1834-4b58-9eff-6c7dec724fec")))
# from a matrix object:
ope <- 253
n.stok <- 7
n.yr <- 8
# somewhat unrealistic: independent returns.
rand.rets <- matrix(rnorm(n.yr * ope * n.stok), ncol = n.stok)
asro <- as.sropt(rand.rets, ope = ope)
rm(rand.rets)
print(asro)

##        SR/sqrt(yr) SRIC/sqrt(yr) 2.5 % 97.5 % T^2 value
## Sharpe        1.05          0.33  0.00   1.43       8.8
##        Pr(>T^2)
## Sharpe     0.27
```

```
# under the alternative, when the mean is nonzero
rand.rets <- matrix(rnorm(n.yr * ope * n.stok, mean = 6e-04,
    sd = 0.01), ncol = n.stok)
asro <- as.sropt(rand.rets, ope = ope)
rm(rand.rets)
print(asro)

##          SR/sqrt(yr) SRIC/sqrt(yr) 2.5 % 97.5 % T^2 value
## Sharpe          2.5           2.1   1.6    3.0         48
##          Pr(>T^2)
## Sharpe  4.6e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# from an xts object
some.rets <- get.rets(c("IBM", "AAPL", "XOM"), from = "2007-01-01",
    to = "2013-01-01")
asro <- as.sropt(some.rets)
print(asro)

##          SR/sqrt(yr) SRIC/sqrt(yr) 2.5 % 97.5 % T^2 value
## Sharpe         0.90          0.53  0.00   1.56        4.8
##          Pr(>T^2)
## Sharpe     0.19
```

One can compute confidence intervals for the population parameter $\zeta_* =_{\mathrm{df}}$ $\sqrt{\boldsymbol{\mu}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}}$, called the 'optimal signal-noise ratio', based on inverting the non-central F-distribution. Estimates of ζ_* can also be computed, via Maximum Likelihood Estimation, or a 'shrinkage' estimate. [7, 17]

```
# confidence intervals:
print(confint(asro, level.lo = 0.05, level.hi = 1))

##      5 % 100 %
## [1,]   0   Inf

# estimation
print(inference(asro, type = "KRS"))

##       [,1]
## [1,] 0.57

print(inference(asro, type = "MLE"))

## [1] 0.64
```

A nice rule of thumb is that, to a first order approximation, the MLE of ζ_* is zero exactly when $\hat{\zeta}_*^2 \leq p/n$, where p is the number of assets. [7, 17] Inspection of this inequality confirms that $\hat{\zeta}_*$ and n can be expressed 'in the same units', meaning that if $\hat{\zeta}_*$ is in $\mathrm{yr}^{-1/2}$, then n should be the number of *years*. For example, if the Markowitz portfolio on 8 assets over 7 years has a Sharpe ratio of $1\mathrm{yr}^{-1/2}$, the MLE will be zero. This can be confirmed empirically as below.

9

```
ope <- 253
zeta.s <- 0.8
n.check <- 1000
df1 <- 10
df2 <- 6 * ope
rvs <- rsropt(n.check, df1, df2, zeta.s, ope, drag = 0)
roll.own <- sropt(z.s = rvs, df1, df2, drag = 0, ope = ope,
    epoch = "yr")
MLEs <- inference(roll.own, type = "MLE")
zerMLE <- MLEs <= 0
crit.value <- 0.5 * (max(rvs[zerMLE])^2 + min(rvs[!zerMLE])^2)
aspect.ratio <- df1/(df2/ope)
cat(sprintf("empirical cutoff for zero MLE is %2.2f yr^{-1}\n",
    crit.value))

## empirical cutoff for zero MLE is 1.68 yr^{-1}

cat(sprintf("the aspect ratio is %2.2f yr^{-1}\n",
    aspect.ratio))

## the aspect ratio is 1.67 yr^{-1}
```

3.1 Approximating Overfit

A high-level sketch of quant work is as follows: construct a trading system with some free parameters, θ, backtest the strategy for $\theta_1, \theta_2, \ldots, \theta_m$, then pick the θ_i that maximizes the Sharpe ratio in backtesting. 'Overfit bias' (variously known as 'datamining bias,' 'garbatrage,' 'backtest arb,' *etc.*) is the upward bias in one's estimate of the true signal-to-noise of the strategy parametrized by θ_{i*} due to one using the same data to select the strategy and estimate it's performance. [2][Chapter 6]

As an example, consider a basic Moving Average Crossover strategy. Here θ is a vector of 2 window lengths. One longs the market exactly when one moving average exceeds the other, and shorts otherwise. One performs a brute force search of all allowable window sizes. Before deciding to deploy money into the MAC strategy parametrized by θ_{i*}, one has to estimate its profitability.

There is a way to roughly estimate overfit bias by viewing the problem as a portfolio problem, and performing inference on the optimal Sharpe ratio. To do this, suppose that $\boldsymbol{x}_i$ is the n-vector of backtested returns associated with θ_i. (I will assume that all the backtests are over the same period of time.) Then approximately embed the backtested returns vectors in the subset of a p-dimensional subspace. That is, by a process like PCA, make the approximation:

$$\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m\} \approx \mathcal{K} \subset \mathcal{L} =_{\mathrm{df}} \{\mathsf{Y}\hat{\boldsymbol{\nu}} \mid \hat{\boldsymbol{\nu}} \in \mathbb{R}^p\}$$

Abusing notation, let $\hat{\zeta}(\theta)$ be the sample Sharpe ratio associated with parameters θ, and also let $\hat{\zeta}(\boldsymbol{x})$ be the Sharpe ratio associated with the vector of returns $\boldsymbol{x}$. Then make the approximation

$$\hat{\zeta}(\theta_*) =_{\mathrm{df}} \max_{\theta_1, \ldots, \theta_m} \hat{\zeta}(\theta_i) \approx \max_{\boldsymbol{x} \in \mathcal{K}} \hat{\zeta}(\boldsymbol{x}) \leq \max_{\boldsymbol{x} \in \mathcal{L}} \hat{\zeta}(\boldsymbol{x}) = \hat{\zeta}_*.$$

This is a conservative approximation: the true maximum over $\mathcal{L}$ is presumably much larger than $\hat{\zeta}(\theta_*)$. One can then use $\hat{\zeta}(\theta_*)$ as $\hat{\zeta}_*$ over a set of p assets, perform inference on ζ_*, which, by a series of approximations as above, is an approximate upper bound on $\zeta(\theta_*)$.

This approximate attack on overfitting will work better when one has a good estimate of p, when m is relatively large and p relatively small, and when the linear approximation to the set of backtested returns is good. Moreover, the definition of $\mathcal{L}$ explicitly allows shorting, whereas the backtested returns vectors $\boldsymbol{x}_i$ may lack the symmetry about zero to make this a good approximation. By way of illustration, consider the case where the trading system is set up such that different θ produce minor variants on a clearly losing strategy: in this case we might have $\hat{\zeta}(\theta_*) < 0$, which cannot hold for $\hat{\zeta}_*$.

One can estimate p via Monte Carlo simulations, by actually performing PCA, or via the 'SWAG' method. Surprisingly, often one's intuitive estimate of the true 'degrees of freedom' in a trading system is reasonably good.

```r
require(TTR)
# brute force search two window MAC
brute.force <- function(lrets, rrets = exp(lrets) -
    1, win1, win2 = win1) {
    mtms <- c(1, exp(cumsum(lrets)))  # prepend a 1.
    # do all the SMAs;
    SMA1 <- sapply(win1, function(n) {
        SMA(mtms, n = n)
    })
    symmetric <- missing(win2)
    if (!symmetric)
        SMA2 <- sapply(win2, function(n) {
            SMA(mtms, n = n)
        })

    mwin <- max(c(win1, win2))
    zeds <- matrix(NaN, nrow = length(win1), ncol = length(win2))
    upb <- if (symmetric)
        length(win1) - 1 else length(win1)
    # 2FIX: vectorize this!
    for (iidx in 1:upb) {
        SM1 <- SMA1[, iidx]
        lob <- if (symmetric)
            iidx + 1 else 1
        for (jidx in lob:length(win2)) {
            SM2 <- if (symmetric)
                SMA1[, jidx] else SMA2[, jidx]
            trades <- sign(SM1 - SM2)
            dum.bt <- trades[mwin:(length(trades) -
                1)] * rrets[mwin:length(rrets)]  # braindead backtest.
            mysr <- as.sr(dum.bt)
            zeds[iidx, jidx] <- mysr$sr
            # abuse symmetry of arithmetic
            # returns
            if (symmetric)
                zeds[jidx, iidx] <- -zeds[iidx, jidx]
        }
```

```
    }
    retv <- max(zeds, na.rm = TRUE)
    return(retv)
}
# simulate one.
sim.one <- function(nbt, win1, ...) {
    lrets <- rnorm(nbt + max(win1), sd = 0.01)
    retv <- brute.force(lrets, win1 = win1, ...)
    return(retv)
}
# set everything up
set.seed(as.integer(charToRaw("e23769f4-94f8-4c36-bca1-28c48c49b4fb")))
ope <- 253
n.yr <- 4
n.obs <- ceiling(ope * n.yr)
LONG.FORM <- FALSE
n.sim <- if (LONG.FORM) 2048 else 1024
win1 <- if (LONG.FORM) c(2, 4, 8, 16, 32, 64, 128,
    256) else c(4, 16, 64, 256)

# run them
system.time(max.zeds <- replicate(n.sim, sim.one(n.obs,
    win1)))

##    user  system elapsed
##   1.383   0.022   1.408

# qqplot;
qqplot(qsropt(ppoints(length(max.zeds)), df1 = 2, df2 = n.obs),
    max.zeds, xlab = "Theoretical Approximate Quantiles",
    ylab = "Sample Quantiles")
qqline(max.zeds, datax = FALSE, distribution = function(p) {
    qsropt(p, df1 = 2, df2 = n.obs)
}, col = 2)
```

Here I illustrate the quality of the approximation for the two-window simple MAC strategy. I generate log returns which are homoskedastic, driftless, and have zero autocorrelation. In this case, *every* MAC strategy has zero expected return (ignoring trading costs). In spite of this deficiency in the market, I find the best combination of window sizes by looking at 4 years of daily data. By selecting the combination of windows with the highest Sharpe ratio, then using that maximal value as an estimate of the selected model's true signal-noise-ratio, I have subjected myself to overfit bias. I repeat this experiment 1024 times, then Q-Q plot the maximal Sharpe ratio values over those experiments versus an optimal Sharpe ratio distribution assuming $p = 2$ in Figure 1. The fit is reasonable[1] except in the case where the maximal in-sample Sharpe ratio is very low (recall that it can be negative for this brute-force search, whereas the optimal Sharpe ratio distribution does not produce negative values). This case is unlikely to lead to a trading catastrophe, however.

It behooves me to replicate the above experiment 'under the alternative,'

---

[1]And much better when the overfitting is more aggressive, which takes more processing time to simulate.
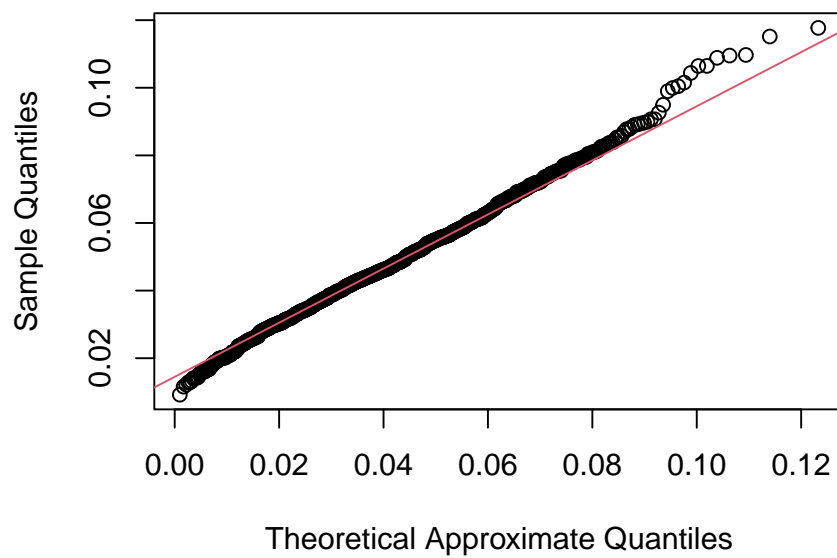
Figure 1: Q-Q plot of 1024 achieved optimal Sharpe ratio values from brute force search over both windows of a Moving Average Crossover under the null of driftless log returns with zero autocorrelation versus the approximation by a 2-parameter optimal Sharpe ratio distribution is shown.

e.g., when the market has autocorrelated returns, to see if the approximation holds up when $\zeta_* > 0$. I leave this for future iterations. Instead, I apply the $p = 2$ approximation to the brute-force MAC overfit on `SPY`.

```
# is MAC on SPY significant?
SPY.lret <- get.rets(c("SPY"), from = "2003-01-01",
    to = "2013-01-01")
# oops! there might be NAs in there!
mysr <- as.sr(SPY.lret, na.rm = TRUE)  # just to get the ope
print(mysr)

##      SR/sqrt(yr) Std. Error t value Pr(>t)
## SPY         0.31       0.32    0.99   0.16

# get rid of NAs
SPY.lret[is.na(SPY.lret)] <- 0
# try a whole lot of windows:
win1 <- seq(4, 204, by = 10)
zeds <- brute.force(SPY.lret, win1 = win1)
SPY.MAC.asro <- sropt(z.s = zeds, df1 = 2, df2 = length(SPY.lret) -
    max(win1), ope = mysr$ope)
print(SPY.MAC.asro)

##          SR/sqrt(yr) SRIC/sqrt(yr) 2.5 % 97.5 % T^2 value
## Sharpe          0.47          0.24  0.00   1.04         2
##          Pr(>T^2)
## Sharpe       0.36

print(inference(SPY.MAC.asro, type = "KRS"))

## [1] 0.33
```

The Sharpe ratio for `SPY` over this period is $0.31\mathrm{yr}^{-1/2}$. The optimal Sharpe ratio for the tested MAC strategies is $0.47\mathrm{yr}^{-1/2}$. The KRS estimate (using the $p = 2$ approximation) for the upper bound on signal-to-noise of the optimal MAC strategy is only $0.33\mathrm{yr}^{-1/2}$. [7] This leaves little room for excitement about MAC strategies on `SPY`.

4   Using the `del_sropt` Class

The class `del_sropt` stores the 'optimal' Sharpe ratio of the optimal hedge-constrained portfolio. See the 'Notes on Sharpe ratio' vignette for more details. There is an object constructor, but likely the **as.del_sropt** function will prove more useful.

```
set.seed(as.integer(charToRaw("364e72ab-1570-43bf-a1c6-ee7481e1c631")))
# from a matrix object:
ope <- 253
n.stok <- 7
n.yr <- 8
# somewhat unrealistic: independent returns,
# under the null
rand.rets <- matrix(rnorm(n.yr * ope * n.stok), ncol = n.stok)
```

```r
# the hedge constraint: hedge out the first
# stock.
G <- diag(n.stok)[1, ]
asro <- as.del_sropt(rand.rets, G, ope = ope)
print(asro)

##        SR/sqrt(yr) 2.5 % 97.5 % F value Pr(>F)
## Sharpe           1     0    1.5     1.4   0.19

# hedge out the first two
G <- diag(n.stok)[1:2, ]
asro <- as.del_sropt(rand.rets, G, ope = ope)
print(asro)

##        SR/sqrt(yr) 2.5 % 97.5 % F value Pr(>F)
## Sharpe        0.99     0    1.5     1.6   0.16

# under the alternative, when the mean is nonzero
rand.rets <- matrix(rnorm(n.yr * ope * n.stok, mean = 6e-04,
    sd = 0.01), ncol = n.stok)
G <- diag(n.stok)[1, ]
asro <- as.del_sropt(rand.rets, G, ope = ope)
print(asro)

##        SR/sqrt(yr) 2.5 % 97.5 % F value  Pr(>F)
## Sharpe         2.6   1.7    3.2     8.8 1.6e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Here I present an example of hedging out `SPY` from a portfolio holding `IBM`, `AAPL`, and `XOM`. The 95% confidence interval on the optimal hedged signal-noise ratio contains zero:

```r
# from an xts object
some.rets <- get.rets(c("SPY", "IBM", "AAPL", "XOM"),
    from = "2007-01-01", to = "2013-01-01")
# without the hedge, allowing SPY position
asro <- as.sropt(some.rets)
print(asro)

##        SR/sqrt(yr) SRIC/sqrt(yr) 2.5 % 97.5 % T^2 value
## Sharpe        1.18          0.76  0.00   1.81       8.4
##        Pr(>T^2)
## Sharpe    0.079 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# hedge out SPY!
G <- diag(dim(some.rets)[2])[1, ]
asro.hej <- as.del_sropt(some.rets, G)
print(asro.hej)

##        SR/sqrt(yr) 2.5 % 97.5 % F value Pr(>F)
## Sharpe         1.1     0    1.7     2.2  0.084 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

15

One can compute confidence intervals for, and perform inference on, the population parameter, $\sqrt{\zeta_{*,\mathsf{I}}^2 - \zeta_{*,\mathsf{G}}^2}$. This is the optimal signal-noise ratio of a hedged portfolio. These are all reported here in the same 'annualized' units of Sharpe ratio, and may be controlled by the `ope` parameter.

```
# confidence intervals:
print(confint(asro, level.lo = 0.05, level.hi = 1))

##      5 % 100 %
## [1,]   0   Inf

# estimation
print(inference(asro, type = "KRS"))

##      [,1]
## [1,] 0.85

print(inference(asro, type = "MLE"))

## [1] 0.92
```

5 Hypothesis Tests

The function `sr_equality_test`, implements the tests of Leung *et al.* and of Wright *et al.* for testing the equality of Sharpe ratio. [10, 19] I have found that these tests can reject the null 'for the wrong reason' when the problem is ill-scaled.

Here I confirm that the tests give approximately uniform p-values under the null in three situations: when stock returns are independent Gaussians, when they are independent and t-distributed, and when they are Gaussian and correlated. Visual confirmation is via Figure 2 through Figure 4, showing Q-Q plots against uniformity of the putative p-values in Monte Carlo simulations with data drawn from the null.

```
# function for Q-Q plot against uniformity
qqunif <- function(x, xlab = "Theoretical Quantiles under Uniformity",
    ylab = NULL, ...) {
    if (is.null(ylab))
        ylab = paste("Sample Quantiles (", deparse(substitute(x)),
            ")", sep = "")
    qqplot(qunif(ppoints(length(x))), x, xlab = xlab,
        ylab = ylab, ...)
    abline(0, 1, col = "red")
}

# under normality.
LONG.FORM <- FALSE
n.ro <- if (LONG.FORM) 253 * 4 else 253 * 2
n.co <- if (LONG.FORM) 20 else 4
n.sim <- if (LONG.FORM) 1024 else 512
set.seed(as.integer(charToRaw("e3709e11-37e0-449b-bcdf-9271fb1666e5")))
```
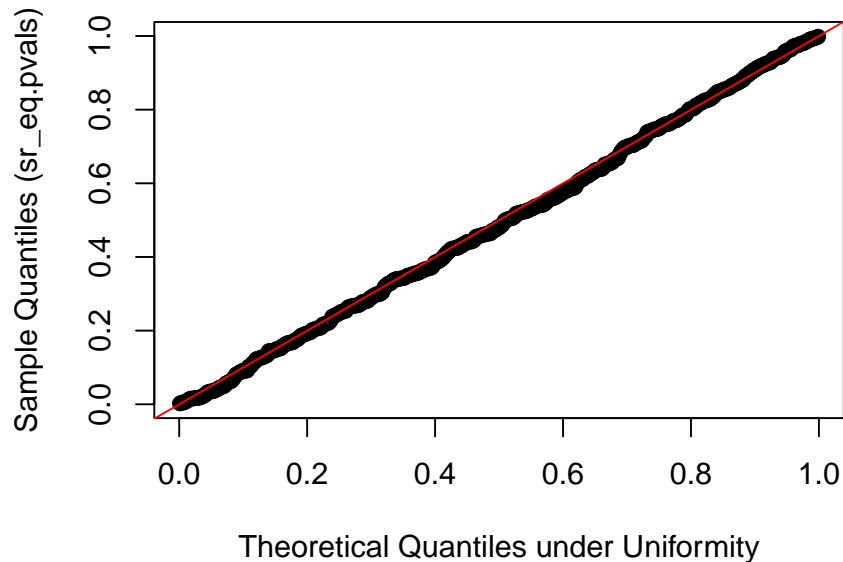
Figure 2: P-values from `sr_equality_test` on 512 multiple realizations of independent, normally distributed returns series are Q-Q plotted against uniformity.

```
afoo <- replicate(n.sim, sr_equality_test(matrix(rnorm(n.ro *
    n.co), ncol = n.co), type = "F"))
sr_eq.pvals <- unlist(afoo["p.value", ])
qqunif(sr_eq.pvals)
```

```
# try heteroskedasticity?
set.seed(as.integer(charToRaw("81c97c5e-7b21-4672-8140-bd01d98d1d2e")))
afoo <- replicate(n.sim, sr_equality_test(matrix(rt(n.ro *
    n.co, df = 4), ncol = n.co), type = "F"))
sr_eq.pvals <- unlist(afoo["p.value", ])
qqunif(sr_eq.pvals)
```

```
# try correlated returns
n.fact <- max(2, n.co - 5)
gen.ret <- function(n1, n2, f = max(2, n2 - 2), fuzz = 0.1) {
    A <- matrix(rnorm(n1 * f), nrow = n1)
    B <- matrix(rnorm(f * n2), nrow = f)
    C <- sqrt(1 - fuzz^2) * A %*% B + fuzz * matrix(rnorm(n1 *
        n2), nrow = n1)
}
set.seed(as.integer(charToRaw("e4d61c2c-efb3-4cba-9a6e-5f5276ce2ded")))
afoo <- replicate(n.sim, sr_equality_test(gen.ret(n.ro,
```
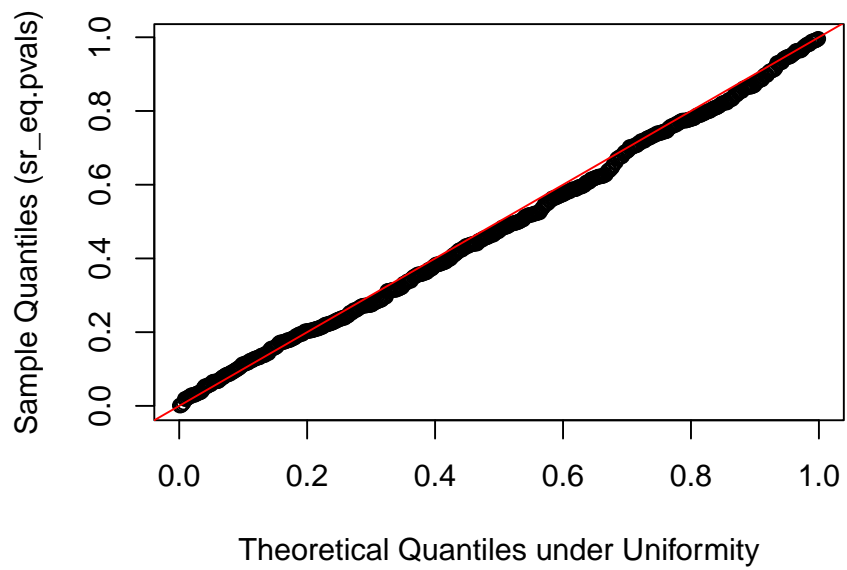
Figure 3: P-values from `sr_equality_test` on 512 multiple realizations of independent, t-distributed returns series are Q-Q plotted against uniformity.
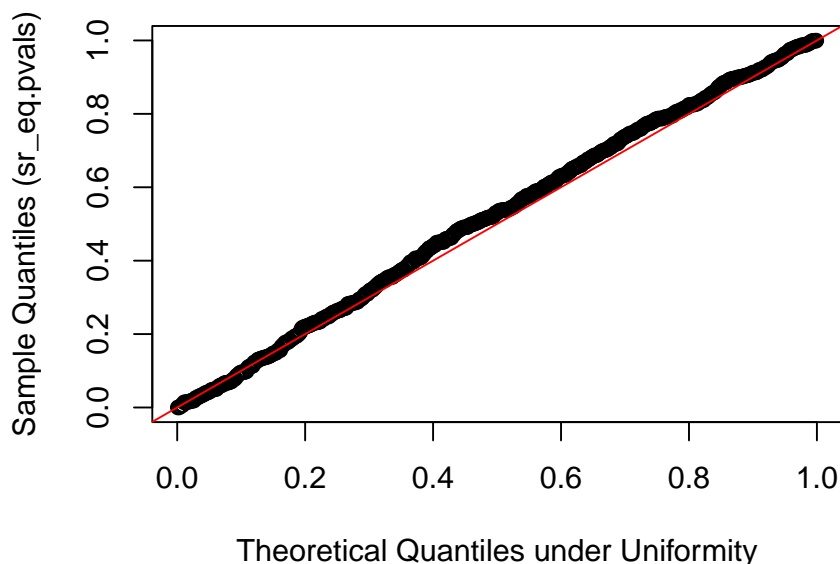
Figure 4: P-values from `sr_equality_test` on 512 multiple realizations of correlated, normally-distributed returns series are Q-Q plotted against uniformity.

```
    n.co, n.fact), type = "F"))
sr_eq.pvals <- unlist(afoo["p.value", ])
qqunif(sr_eq.pvals)
```

As a followup, I consider the case where the returns are those of several randomly generated market-timing strategies which are always fully invested long or short in the market, with equal probability. I look at 128 simulations of 50 random market timing strategies rebalancing weekly on a 10 year history of `SPY`. The 'p-values' from `sr_equality_test` applied to the returns are Q-Q plotted against uniformity in Figure 5. The type I rate of this test is far larger than the nominal rate: it is rejecting for 'uninteresting reasons'. I suspect this test breaks down because of small sample size or small 'aspect ratio' (ratio of weeks to strategies in this example). In summary, one should exercise caution when interpreting the results of the Sharpe equality test: it would be worthwhile to compare the results against a 'Monte Carlo null' as illustrated here.

```
n.co <- if (LONG.FORM) 100 else 50
n.sim <- if (LONG.FORM) 1024 else 128
SPY.lret <- get.rets(c("SPY"), from = "2003-01-01",
    to = "2013-01-01")
SPY.wk.rret <- my.periodReturn(lr2mtm(SPY.lret), period = "weekly",
    type = "arithmetic")
gen.tim <- function(n2) {
```
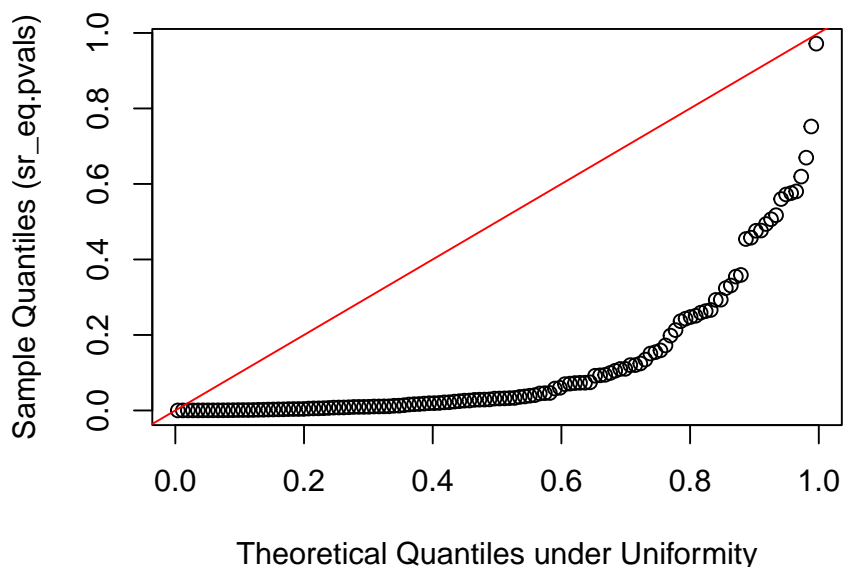
Figure 5: P-values from `sr_equality_test` on 128 multiple realizations of 50 market timing strategies' returns series are Q-Q plotted against uniformity. Nominal coverage is not maintained: the test is far too liberal in this case.

```
    mkt.timing.signal <- sign(rnorm(n2 * length(SPY.wk.rret)))
    mkt.ret <- matrix(rep(SPY.wk.rret, n2) * mkt.timing.signal,
        ncol = n2)
}
set.seed(as.integer(charToRaw("447cfe85-b612-4b14-bd01-404e6e99aca4")))
system.time(afoo <- replicate(n.sim, sr_equality_test(gen.tim(n.co),
    type = "F")))

##    user  system elapsed
##   1.175   0.012   1.187

sr_eq.pvals <- unlist(afoo["p.value", ])
qqunif(sr_eq.pvals)
```

5.1   Equality Test with Fancy Covariance

The `sr_equality_test` function now accepts an optional function to perform the inner variance-covariance estimation. I do not think this will correct the problems noted previously for the ill-scaled case. It does, however, allow one to take into account *e.g.*, heteroskedasticity and autocorrelation, as follows. Note that the use of a 'fancy' covariance estimator does not change the conclusions

20

of the Sharpe equality test in this case.

```r
# get returns
some.rets <- as.matrix(get.rets(c("IBM", "AAPL", "XOM"),
    from = "2007-01-01", to = "2013-01-01"))
# using the default vcov
test.vanilla <- sr_equality_test(some.rets, type = "F")
print(test.vanilla)

##
##  test for equality of Sharpe ratio, via F test
##
## data:  some.rets
## T2 = 3, contrasts = 2, p-value = 0.3
## alternative hypothesis: true sum squared contrasts of SNR is not equal to 0

if (require(sandwich)) {
    # and a fancy one:
    test.HAC <- sr_equality_test(some.rets, type = "F",
        vcov.func = vcovHAC)
    print(test.HAC)
}

##
##  test for equality of Sharpe ratio, via F test
##
## data:  some.rets
## T2 = 3, contrasts = 2, p-value = 0.3
## alternative hypothesis: true sum squared contrasts of SNR is not equal to 0
```

6 Asymptotics

6.1 Variance Covariance of the Sharpe ratio

Given n observations of the returns of p assets, the covariance of the sample Sharpe ratio can be estimated via the Delta method. This operates by stacking the p vector of returns on top of the p vector of the returns squared element-wise. One then estimates the covariance of this $2p$ vector, using one's favorite covariance estimator. This estimate is then translated back to an estimate of the covariance of the p vector of Sharpe ratio values. This process was described by Lo, and Ledoit and Wolf for the $p = 1$ case, and is used in the Sharpe equality tests of Leung *et al.* and of Wright *et al.*. [11, 9, 10, 19]

This basic function is available via the `sr_vcov` function, which allows one to pass in a function to perform the covariance estimate of the $2p$ vector. The passed in function should operate on an `lm` object. The default is the `vcov` function; other sensible choices include `sandwich:vcovHC` and `sandwich:vcovHAC` to deal with heteroskedasticity and autocorrelation.

```r
# get returns
some.rets <- as.matrix(get.rets(c("IBM", "AAPL", "XOM"),
    from = "2007-01-01", to = "2013-01-01"))
```

```
ope <- 252
vanilla.Sig <- sr_vcov(some.rets, ope = ope)
print(vanilla.Sig)

## $SR
##       [,1]
## IBM  0.54
## XOM  0.15
## AAPL 0.83
##
## $Ohat
##          IBM   XOM  AAPL
## IBM   0.167 0.098 0.092
## XOM   0.098 0.167 0.076
## AAPL  0.092 0.076 0.171
##
## $p
## [1] 3

if (require(sandwich)) {
    HC.Sig <- sr_vcov(some.rets, vcov = vcovHC, ope = ope)
    print(HC.Sig$Ohat)
}

##          IBM   XOM  AAPL
## IBM   0.167 0.099 0.092
## XOM   0.099 0.167 0.076
## AAPL  0.092 0.076 0.171

if (require(sandwich)) {
    HAC.Sig <- sr_vcov(some.rets, vcov = vcovHAC, ope = ope)
    print(HAC.Sig$Ohat)
}

##          IBM   XOM  AAPL
## IBM   0.161 0.069 0.090
## XOM   0.069 0.111 0.049
## AAPL  0.090 0.049 0.166
```

6.2   Variance Covariance of the Markowitz Portfolio

Given a vector of returns, $\boldsymbol{x}$, prepending a one and taking the uncentered moment gives a 'unified' parameter:

$$\Theta =_{\mathrm{df}} \mathrm{E}\left[\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^{\top}\right] = \begin{bmatrix} 1 & \boldsymbol{\mu}^{\top} \\ \boldsymbol{\mu} & \Sigma + \boldsymbol{\mu}\boldsymbol{\mu}^{\top} \end{bmatrix},$$

where $\tilde{\boldsymbol{x}} = \left[1, \boldsymbol{x}^{\top}\right]^{\top}$. The inverse of Θ contains the (negative) Markowitz portfolio:

$$\Theta^{-1} = \begin{bmatrix} 1 + \boldsymbol{\mu}^{\top}\Sigma^{-1}\boldsymbol{\mu} & -\boldsymbol{\mu}^{\top}\Sigma^{-1} \\ -\Sigma^{-1}\boldsymbol{\mu} & \Sigma^{-1} \end{bmatrix} = \begin{bmatrix} 1 + \zeta_*^2 & -\boldsymbol{\nu}_*^{\top} \\ -\boldsymbol{\nu}_* & \Sigma^{-1} \end{bmatrix}.$$

These computations hold for the sample estimator $\hat{\Theta} =_{\mathrm{df}} \frac{1}{n}\tilde{\mathsf{X}}^{\top}\tilde{\mathsf{X}}$, where $\tilde{\mathsf{X}}$ is the matrix whose rows are the n vectors $\tilde{\boldsymbol{x}}_i^{\top}$.

By the multivariate Central Limit Theorem, [18]

$$\sqrt{n}\left(\text{vech}\left(\hat{\Theta}\right) - \text{vech}\left(\Theta\right)\right) \rightsquigarrow \mathcal{N}\left(0, \Omega\right),$$

for some Ω which can be estimated from the data. Via the delta method the asymptotic distribution of vech $\left(\hat{\Theta}^{-1}\right)$, which contains $-\hat{\Sigma}^{-1}\hat{\mu}$, can be found. See the other vignette for more details.

This functionality is available via the `ism_vcov` function. This function returns the sample estimate and estimated asymptotic variance of vech $\left(\hat{\Theta}^{-1}\right)$ (with the leading one removed). This function also allows one to pass in a callback to perform the covariance estimation. The default is the `vcov` function; other sensible choices include `sandwich:vcovHC` and `sandwich:vcovHAC`.

```r
# get returns
some.rets <- get.rets(c("IBM", "AAPL", "XOM"), from = "2007-01-01",
    to = "2013-01-01")

ism.wald <- function(X, vcov.func = vcov) {
    # negating returns is idiomatic to get +
    # Markowitz
    ism <- ism_vcov(-as.matrix(X), vcov.func = vcov.func)
    ism.mu <- ism$mu[1:ism$p]
    ism.Sg <- ism$Ohat[1:ism$p, 1:ism$p]
    retval <- ism.mu/sqrt(diag(ism.Sg))
    dim(retval) <- c(ism$p, 1)
    rownames(retval) <- rownames(ism$mu)[1:ism$p]
    return(retval)
}

wald.stats <- ism.wald(some.rets)
print(t(wald.stats))

##      IBM   XOM AAPL
## [1,] 0.59 -0.83  1.7

if (require(sandwich)) {
    wald.stats <- ism.wald(some.rets, vcov.func = sandwich::vcovHAC)
    print(t(wald.stats))
}

##      IBM   XOM AAPL
## [1,] 0.58 -0.91  1.7
```

7 Miscellanea

7.1 Distribution Functions

There are `dpqr` functions for the Sharpe ratio distribution, as well as the 'optimal Sharpe ratio' distribution. These are merely rescaled non-central t and F distributions, provided for convenience, and for testing purposes. See the help for `dsr` and `dsropt` for more details.

References

[1] M. Andrecut. Portfolio optimization in R, 2013. URL http://arxiv.org/abs/1307.0450.

[2] David R. Aronson. *Evidence-Based Technical Analysis: Applying the Scientific Method and Statistical Inference to Trading Signals*. Wiley, Hoboken, NJ, November 2006. URL http://www.evidencebasedta.com/.

[3] David H. Bailey and Marcos Lopez de Prado. The Sharpe Ratio Efficient Frontier. *Social Science Research Network Working Paper Series*, April 2011. URL http://ssrn.com/abstract=1821643.

[4] Ronald Aylmer Fisher. Applications of "Student's" distribution. *Metron*, 5:90–104, 1925. URL http://hdl.handle.net/2440/15187.

[5] William Sealy Gosset. The probable error of a mean. *Biometrika*, 6(1):1–25, March 1908. URL http://dx.doi.org/10.2307/2331554. Originally published under the pseudonym "Student".

[6] N. L. Johnson and B. L. Welch. Applications of the non-central t-distribution. *Biometrika*, 31(3-4):362–389, March 1940. doi: 10.1093/biomet/31.3-4.362. URL http://dx.doi.org/10.1093/biomet/31.3-4.362.

[7] Tatsuya Kubokawa, C. P. Robert, and A. K. Md. E. Saleh. Estimation of noncentrality parameters. *Canadian Journal of Statistics*, 21(1):45–57, 1993. URL http://www.jstor.org/stable/3315657.

[8] Bruno Lecoutre. Another look at confidence intervals for the noncentral T distribution. *Journal of Modern Applied Statistical Methods*, 6(1):107–116, 2007. URL http://www.univ-rouen.fr/LMRS/Persopage/Lecoutre/telechargements/Lecoutre_Another_look-JMSAM2007_6(1).pdf.

[9] Olivier Ledoit and Michael Wolf. Robust performance hypothesis testing with the Sharpe ratio. *Journal of Empirical Finance*, 15(5):850–859, Dec 2008. ISSN 0927-5398. doi: http://dx.doi.org/10.1016/j.jempfin.2008.03.002. URL http://www.ledoit.net/jef2008_abstract.htm.

[10] Pui-Lam Leung and Wing-Keung Wong. On testing the equality of multiple Sharpe ratios, with application on the evaluation of iShares. *Journal of Risk*, 10(3):15–30, 2008. URL http://www.risk.net/digital_assets/4760/v10n3a2.pdf.

[11] Andrew W. Lo. The Statistics of Sharpe Ratios. *Financial Analysts Journal*, 58(4), July/August 2002. URL http://ssrn.com/paper=377260.

[12] Thomas Lumley, Paula Diehr, Scott Emerson, and Lu Chen. The importance of the normality assumption in large public health data sets. *Annu Rev Public Health*, 23:151–69+, 2002. ISSN 0163-7525. doi: 10.1146/annurev.publhealth.23.100901.140546. URL http://arjournals.annualreviews.org/doi/pdf/10.1146/annurev.publhealth.23.100901.140546.

[13] Robert E. Miller and Adam K. Gehr. Sample size bias and Sharpe's performance measure: A note. *Journal of Financial and Quantitative Analysis*, 13(05):943–946, 1978. doi: 10.2307/2330636. URL `http://dx.doi.org/10.1017/S0022109000014216`.

[14] J. D. Opdyke. Comparing Sharpe ratios: So where are the p-values? *Journal of Asset Management*, 8(5), 2007. URL `http://ssrn.com/abstract=886728`.

[15] Steven E. Pav. A short sharpe course. Privately Published, 2017. URL `https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3036276`.

[16] William F. Sharpe. Mutual fund performance. *Journal of Business*, 39:119, 1965. URL `http://ideas.repec.org/a/ucp/jnlbus/v39y1965p119.html`.

[17] M. C. Spruill. Computation of the maximum likelihood estimate of a noncentrality parameter. *Journal of Multivariate Analysis*, 18(2):216–224, 1986. ISSN 0047-259X. doi: 10.1016/0047-259X(86)90070-9. URL `http://www.sciencedirect.com/science/article/pii/0047259X86900709`.

[18] Larry Wasserman. *All of Statistics: A Concise Course in Statistical Inference*. Springer Texts in Statistics. Springer, 2004. ISBN 9780387402727. URL `http://books.google.com/books?id=th3fbFI1DaMC`.

[19] John Alexander Wright, Sheung Chi Phillip Yam, and Siu Pang Yung. A note on a test for the equality of multiple Sharpe ratios and its application on the evaluation of iShares. Technical report, 2012. URL `http://www.sta.cuhk.edu.hk/scpy/Preprints/John%20Wright/A%20test%20for%20the%20equality%20of%20multiple%20Sharpe%20ratios.pdf`. to appear.