# PeerPerformance: Luck-Corrected Peer Performance Analysis for Hedge Funds

*by David Ardia and Benjamin Seguin*

**Abstract** The **PeerPerformance** package provides comprehensive tools for peer-performance evaluation of financial investments with explicit correction for luck and false discoveries. This vignette demonstrates the package's core functionality for alpha-based analysis, Sharpe ratio testing, and modified Sharpe ratio evaluation, while explaining the underlying methodology that addresses the multiple comparison problem inherent in fund performance evaluation. The package implements peer performance ratios that measure the percentage of peers a focal fund outperforms and underperforms, after correction for luck using false discovery rate methods.

## 1 Introduction

Evaluating the performance of investment funds relative to their peers is challenging when many funds have similar true performance. A naive approach ranks funds by estimated performance (e.g., alpha or Sharpe ratio) and classifies a fund as 'outperforming' or 'underperforming' based on its percentile rank. However, this ignores the possibility that multiple funds have equal true performance, leading to false discoveries of outperformance or underperformance. In the extreme case where all funds truly have the same performance, a percentile ranking still assigns a random outperformance percentage between 0 and 100% to each fund, conveying no real information. Moreover, investors tend to reallocate capital to funds labeled as top performers, so falsely identifying a fund as an outperformer (when its peers are actually equal performers) can result in inefficient capital allocation.

To address these issues, Ardia and Boudt (2018) propose a triple-layered peer performance framework that explicitly allows for equal performance among peer funds. In this framework, each fund $i$ is characterized by three "peer performance ratios": an equal-performance ratio ($\pi_i^0$), an outperformance ratio ($\pi_i^+$), and an underperformance ratio ($\pi_i^-$). These parameters are defined as the true proportion of funds in the peer group that perform equally well as fund $i$, that are outperformed by fund $i$, and that outperform fund $i$, respectively. By definition, $\pi_i^0 + \pi_i^+ + \pi_i^- = 1$ for each fund $i$. They show that evaluating funds on this triple scale (instead of a single rank) provides a more informative and false-discovery–controlled assessment of performance.

Estimating the peer performance ratios is non-trivial because it requires determining which performance differences are statistically significant among a multitude of fund pairs. Ardia and Boudt (2018) develop a non-parametric estimation procedure for $(\pi_i^0, \pi_i^+, \pi_i^-)$ that controls for multiple testing false positives. In an empirical study on hedge funds, they find that this luck-correction leads to markedly different conclusions than naive rankings: for example, a portfolio of funds selected for highest outperformance ratios achieves better risk-adjusted returns than one selected by highest estimated alpha. The open-source **PeerPerformance** R package implements all the methodology from their paper, making it easy for practitioners to compute luck-adjusted peer performance measures.

In addition to peer performance ratios, the **PeerPerformance** package also provides tools for rigorous comparison of funds' risk-adjusted returns through Sharpe ratios. Classic Sharpe ratio comparisons assume normally distributed returns and can be misleading when returns are non-normal (as is common for hedge funds). Ardia and Boudt (2015) develop a statistical test for the equality of modified Sharpe ratios (mSR) of two investments. The modified Sharpe ratio (Favre and Galeano, 2002; Gregoriou and Gueyie, 2003) replaces the standard deviation with the modified Value-at-Risk, incorporating skewness and kurtosis of returns. Ardia and Boudt's test accounts for non-normal return distributions and overlapping data using a bootstrap approach, and they demonstrate its good size and power in simulations. The **PeerPerformance** package implements this testing framework for both the Sharpe ratio (using the Ledoit and Wolf (2008) approach) and the modified Sharpe ratio, allowing users to formally test if one fund's risk-adjusted performance is significantly different from another's.

In this vignette, we present the theoretical background of the peer performance ratios and demonstrate the functionality of the **PeerPerformance** package. We begin by outlining the methodology for estimating equal-, out-, and underperformance ratios using multiple hypothesis testing with false discovery rate (FDR) control. We then describe the implementation in R, and provide examples using the package's built-in hedge fund dataset (`hfdata`). We also highlight how to use the package's tools for Sharpe ratio comparisons.

## 2 Methodology

### 2.1 Peer Performance Ratio Framework

Consider a peer group of $N$ funds plus a focal fund $i$ (so $N$ peers in the universe aside from $i$). Let $\Delta_{i,j}$ denote the true difference in performance between fund $i$ and fund $j$ (we write $j \neq i$ for a peer fund). Following Ardia and Boudt (2018), performance is measured in terms of risk-adjusted expected return, typically the intercept (alpha) from a factor model regression. For example, one may estimate for each pair $(i, j)$ the regression:

$$(r_{i,t} - r_{j,t}) = \alpha_{i-j} + \boldsymbol{\beta}_{i-j}^{\top} \mathbf{f}_t + \varepsilon_{i-j,t},$$

where $r_{i,t}$ and $r_{j,t}$ are the net returns of funds $i$ and $j$ at time $t$, $\mathbf{f}t$ is a vector of common risk factors (e.g., market and style factors), $\alpha_{i-j} = \Delta_{i,j}$ is the difference in their alphas (risk-adjusted excess returns), and $\boldsymbol{\beta}_{i-j}$ captures any differential factor exposures. The null hypothesis of equal performance between $i$ and $j$ is $H_0 : \Delta_{i,j} = 0$, i.e. $\alpha_{i-j} = 0$. The alternative $H_1 : \Delta_{i,j} \neq 0$ means one fund truly outperforms the other.

By Ardia and Boudt's definitions, the equal-performance ratio $\pi_i^0$ is the proportion of peer funds $j$ for which $\Delta_{i,j} = 0$ (fund $i$ has truly the same performance as $j$). The outperformance ratio $\pi_i^+$ is the proportion of peers for which $\Delta_{i,j} > 0$ (fund $i$ is truly better), and the underperformance ratio $\pi_i^-$ is the proportion for which $\Delta_{i,j} < 0$ (fund $i$ is truly worse). These ratios sum to 1: $\pi_i^0 + \pi_i^+ + \pi_i^- = 1$. In practice, of course, we do not know the true $\Delta_{i,j}$ values and must infer these ratios from observed returns.

**Multiple Testing Approach**: The key insight of Ardia and Boudt (2018) is that we can estimate $(\pi_i^0, \pi_i^+, \pi_i^-)$ by performing pairwise significance tests for all fund comparisons and then aggregating the results while controlling for false discoveries. For a fixed focal fund $i$, consider the $N$ hypothesis tests $H_0 : \Delta_{i,j} = 0$ vs $H_1 : \Delta_{i,j} \neq 0$ for each peer $j = 1, \ldots, N$. From each test we obtain a p-value $p_{i,j}$ for the null of equal performance. If $p_{i,j}$ is very small, we have evidence that fund $i$'s performance differs significantly from fund $j$'s. However, with $N$ tests, some p-values are small by chance even if many peers have $\Delta_{i,j} = 0$. Therefore, instead of naively counting how many $p_{i,j}$ fall below a fixed significance level, we employ a false discovery rate approach (Storey, 2002) to estimate the fraction of true null hypotheses.

Denote by $n = N$ the number of peer funds. Under the assumption that p-values are uniformly distributed on $[0, 1]$ whenever the null is true, the proportion of true nulls (equal performers) can be inferred from the tail of the p-value distribution. In particular, if $\pi_i^0$ is the true proportion of equal-performance peers, about $\pi_i^0 \times n$ of the p-values $p_{i,j}$ should be uniformly distributed (for true nulls) and thus large p-values predominantly correspond to those nulls. Let $\lambda$ be a high threshold (e.g. $\lambda = 0.5$ or 0.6). The expected number of p-values exceeding $\lambda$ is

$$(1 - \lambda) \cdot n \cdot \pi_i^0$$

since under true nulls $\mathbb{P}(p > \lambda) = 1 - \lambda$, and false nulls contribute negligibly to that extreme. Therefore, a natural estimator for the number of equal-performance peers is:

$$\hat{n}_i^0 = \min \left\{ \frac{\sum_{j \neq i} \mathbf{1}\{p_{i,j} \geq \lambda\}}{1 - \lambda},\ n \right\},$$

where the indicator $\mathbf{1}\{p_{i,j} \geq \lambda\}$ counts how many p-values are above $\lambda$. The result is capped at $n$ since $\pi_i^0$ cannot exceed 100%. Ardia and Boudt implement a small bias correction factor $c_i^0$ (via a truncation adjustment and jackknife, see Appendix A of their paper) to improve this estimator's accuracy, but the core idea remains the same. Dividing by $n$ yields the estimated equal-performance ratio:

$$\hat{\pi}_i^0 = \frac{\hat{n}_i^0}{n},$$

This estimator $\hat{\pi}_i^0$ is essentially a sample average of the indicator $\mathbf{1}\{p_{i,j} \geq \lambda\}$, scaled up by $(1 - \lambda)^{-1}$. Under suitable regularity conditions (in particular, a large number of independent or weakly dependent tests), $\hat{\pi}_i^0$is unbiased and consistent for the true $\pi_i^0$. In practice, one must choose the threshold $\lambda$. A larger $\lambda$ makes it more likely that the p-values above $\lambda$ truly correspond to equal-performance cases, but using too high a threshold leaves few observations and increases variance of the estimator. Ardia and Boudt (2018) recommend a data-driven choice of $\lambda$ by minimizing the estimated mean squared error of $\hat{\pi}_i^0$. In their hedge fund application, the optimal $\lambda_i$ varied between 0.30 and

0.70 across funds. By default, the **PeerPerformance** package automatically selects an appropriate $\lambda$ for each fund via this procedure.

Once $\hat{\pi}_i^0$ (the proportion of equal performers) is obtained, the remaining proportion $1 - \hat{\pi}_i^0$ corresponds to fund $i$'s unequal peers, i.e. those for which $\Delta_{i,j} \neq 0$. This remaining mass must be split between $\hat{\pi}_i^+$ (outperformance) and $\hat{\pi}_i^-$ (underperformance). Intuitively, $\hat{\pi}_i^+$ should be based on the fraction of peers for which fund $i$ appears to significantly outperform $j$, adjusted for expected false positives, and similarly for $\hat{\pi}_i^-$ with roles reversed. Ardia and Boudt achieve this by selecting an upper tail significance cutoff $\gamma^+$ (e.g. 5% or 10%) for detecting positive performance differentials and a lower tail cutoff $\gamma^-$ for negative differentials (e.g. $\gamma^- = 0.05$ corresponds to 95% upper percentile for a one-sided test). Let $\hat{q}^{\gamma^+}$ be the $\gamma^+$quantile of the $p - values$ (or an equivalent significance threshold) for fund $i$. Essentially, $\hat{q}^{\gamma^+}$ is chosen such that about $\gamma^+ \times n$ of the smallest p-values are considered significant in the positive direction. Then the estimated outperformance ratio is defined as:

$$\hat{\pi}_i^+ = \frac{1}{n} \max \left\{ \#\left\{ j : p_{i,j} \leq \hat{q}^{\gamma^+} \text{ and } \hat{\Delta}_{i,j} > 0 \right\} - \hat{n}_i^0 (1 - \gamma^+),\ 0 \right\}.$$

In words, $\#\{j : p_{i,j} \leq \hat{q}^{\gamma^+} \text{ and } \hat{\Delta}_{i,j} > 0\}$ is the number of peer funds that fund $i$ appears to significantly outperform (i.e. those with very low p-values and with $i$'s estimated alpha higher than $j$'s). From this we subtract $\hat{n}_i^0(1 - \gamma^+)$, which is the expected number of false positives among those discoveries (since $\hat{n}_i^0$ of the tests are null, a fraction $1 - \gamma^+$ of those null p-values would fall below the $\gamma^+$ threshold by random chance). The $\max\{\cdot, 0\}$ ensures we don't get a negative estimate. Similarly, letting $\hat{q}^{\gamma^-}$ be a significance threshold for the lower tail, the estimated underperformance ratio is:

$$\hat{\pi}_i^- = \frac{1}{n} \max \left\{ \#\left\{ j : p_{i,j} \leq \hat{q}^{\gamma^-} \text{ and } \hat{\Delta}_{i,j} < 0 \right\} - \hat{n}_i^0 \cdot \gamma^-,\ 0 \right\},$$

Here $\#\{j : p_{i,j} \leq \hat{q}^{\gamma^-} \text{ and } \hat{\Delta}_{i,j} < 0\}$ counts the cases where fund $i$ looks significantly worse than peer $j$ (very low p-value and $i$'s performance estimate is lower), and $\hat{n}_i^0 \cdot \gamma^-$ is the expected number of false discoveries in that lower tail (since for $\hat{n}_i^0$ true nulls, a fraction $\gamma^-$ would produce p-values in the lowest $\gamma^-$tail by chance). In practice one might use $\gamma^+ = \gamma^-$ (for example 0.05 for both, corresponding to 5% significance in each tail), or slightly asymmetric values if focusing on one tail. In Ardia and Boudt's simulations they consider both 10% and 5% significance levels for illustration.

By construction, $\hat{\pi}_i^0 + \hat{\pi}_i^+ + \hat{\pi}_i^- = 1$ for each fund (any slight imbalance is corrected by the way the tail thresholds are defined). The result is a triple $(\hat{\pi}_i^0, \hat{\pi}_i^+, \hat{\pi}_i^-)$ that summarizes fund $i$'s relative performance in a distributional sense: for example, if $\hat{\pi}_i^+ = 0.20$ and $\hat{\pi}_i^- = 0.10$, we interpret that fund $i$ is significantly better than 20% of its peers and significantly worse than 10% of its peers, while the remaining 70% of peers are statistically indistinguishable (equal performance). This provides more nuance than a simple rank. In particular, a high outperformance ratio coupled with a high equal-performance ratio would indicate that while fund $i$ is near the top of the group, many peers are statistically on par with it; conversely, a moderate outperformance ratio with very low equal-performance ratio would indicate that all differences are clear-cut (either $i$ beats them or loses to them).

**Comparison to Alternative Estimates:** The peer performance ratios generalize and improve upon simpler peer-ranking metrics. If one naively assumed no equal-performance cases ($\pi_i^0 = 0$ for all $i$), then each fund's outperformance could be estimated by its percentile rank among peers. Ardia and Boudt show that this percentile–rank approach systematically overestimates $\pi_i^+$ and $\pi_i^-$ when funds have similar abilities. In their simulation study, a scenario where all funds truly had equal performance yielded random percentile-based outperformance rates (mean 50%) even though the true $\pi_i^+$ were zero; in more realistic mixed-performance scenarios, the rank approach inflated the outperformance of top-ranked funds and the underperformance of bottom-ranked funds by substantial margins. They also compare to a significance test-counting approach, which would classify a peer as outperformed by fund $i$ if the pairwise test is significant at some level (say 5%) and $\hat{\Delta}_{i,j} > 0$. This approach is an improvement over pure ranking because it acknowledges equality when a difference is statistically insignificant. However, using a fixed significance cutoff for each test separately still misses the multiplicity issue — it tends to underestimate $\pi_i^0$ (equal performers) and thus slightly overestimate $\pi_i^+$ and $\pi_i^-$ when many tests are performed. By contrast, the proposed FDR-corrected estimators explicitly subtract the expected number of false positives, yielding approximately unbiased estimates of the true ratios. Monte Carlo simulations in the 2018 paper show that the bias of $\hat{\pi}_i^0$ and $\hat{\pi}_i^+$ is negligible across various scenarios, whereas the percentile-rank estimator for $\pi^+$ had an upward bias around 0.30 (30 percentage points) in their tests. In summary, the triple $(\hat{\pi}_i^0, \hat{\pi}_i^+, \hat{\pi}_i^-)$ provides a luck-adjusted performance profile for each fund, robustly accounting for the possibility that many funds may be statistically indistinguishable from each other.

### 2.2 Sharpe and Modified Sharpe Ratio Testing

Apart from peer group comparisons, performance analysts often want to directly compare two funds or portfolios on a traditional risk-adjusted metric like the Sharpe ratio. The Sharpe ratio is the ratio of a strategy's mean excess return to its return standard deviation, and is a standard measure of risk-adjusted performance. However, hedge fund returns often violate the normal distribution assumption underlying Sharpe ratio significance tests. They exhibit skewness, fat tails, and other nonlinearities that make the Sharpe ratio less informative (investors may care about higher moments). To address this, Favre and Galeano (2002) and Gregoriou and Gueyie (2003) introduced the modified Sharpe ratio (mSR), defined as the ratio of the mean excess return to the Modified Value-at-Risk (Modified VaR) at a given confidence level. The Modified VaR is computed via the Cornish–Fisher expansion using the asset's sample skewness and kurtosis, which adjusts the VaR for non-normality. Intuitively, mSR penalizes downside risk accounting for asymmetry and tail risk, thus providing a more appropriate performance measure for non-normal returns.

Ardia and Boudt (2015) develop a hypothesis test for comparing the modified Sharpe ratios of two investments. Given two return series, $x_t$ and $y_t$, with modified Sharpe ratios $\text{mSR}_x$ and $\text{mSR}_y$, the null hypothesis is $H_0 : \text{mSR}_x = \text{mSR}_y$ (equal risk-adjusted performance) and the alternative is that one is higher. They derive an asymptotic test statistic exploiting the fact that the modified Sharpe ratio, under certain conditions, is asymptotically normal with a known standard error formula. For finite samples, especially when returns are autocorrelated or have other time-series effects, they advocate a bootstrap approach to obtain the p-value for the test. This involves computing the studentized test statistic on many resampled return series (preserving the dependence structure via moving block bootstrap, for instance) to approximate its null distribution (Ledoit and Wolf (2008) use a similar bootstrap for Sharpe ratio comparisons). Ardia and Boudt's simulation results confirm that the test has good size (type I error close to nominal) and power to detect differences in modified Sharpe ratios, whereas a naive test ignoring non-normality could be mis-calibrated. In an empirical example with hedge fund returns, they found that relying on the regular Sharpe ratio alone would have missed significant differences that the modified Sharpe ratio test identified.

The **PeerPerformance** package implements these comparison tests as well as screening tools analogous to the peer performance ratios but based on Sharpe measures. In particular:

• `sharpeTesting(x, y)` performs a hypothesis test of equal Sharpe ratio between two return series x and y. It can use either an asymptotic approximation or a bootstrap method as in Ledoit and Wolf (2008) for improved robustness. The null hypothesis is $H_0 : SR_x = SR_y$, and the function returns a test statistic and p-value.

• `msharpeTesting(x, y)` similarly tests for equality of modified Sharpe ratios of two funds (as described in Ardia and Boudt (2015)). By default it may use the asymptotic test (which is valid for reasonably large samples under mild conditions), and options are provided to use a bootstrap for exact inference. The output includes a p-value indicating whether the difference in modified Sharpe is statistically significant.

• `sharpeScreening` and `msharpeScreening` are analogous to `alphaScreening` (peer performance analysis based on alpha) but instead use Sharpe or modified Sharpe as the performance metric.

These functions compute pairwise comparisons of Sharpe ratios between all pairs of funds in a group, and then estimate each fund's Sharpe outperformance ratio (the percentage of peers it has a higher Sharpe than, accounting for luck) in a similar triple $(\pi^0, \pi^+, \pi^-)$ fashion. Internally, `sharpeScreening` uses the multiple-comparison bootstrap test from Ledoit and Wolf (2008) for each fund pair, while `msharpeScreening` uses the modified Sharpe test. Because these involve potentially many heavy computations (bootstrapping each of $N(N-1)/2$ pairs), the functions support parallel execution. In summary, the methodology underlying **PeerPerformance** combines robust hypothesis testing with multiple comparisons adjustments to provide luck-corrected performance measures. Whether one is interested in pairwise fund comparisons (via Sharpe or modified Sharpe tests) or a peer group evaluation of a single fund (via outperformance ratios), the package leverages the approaches of Ardia and Boudt (2018) and Ardia and Boudt (2015) to deliver statistically rigorous results.

## 3 Implementation in R

The **PeerPerformance** package (citation) provides a user-friendly interface to apply the above methods in R. The package functions are designed to handle a universe of funds' returns and compute the desired performance statistics, taking care of parallelization and underlying test calculations. In this section, we describe the key functions and demonstrate their usage with the built-in dataset.

### 3.1 Data Format

The primary dataset provided by the package is `hfdata`, which contains monthly returns for 100 hedge funds over 60 months. While anonymized and stylized for confidentiality, it reflects realistic fund dynamics and is designed to showcase the functionalities of the **PeerPerformance** package. Since no time index is included, users may assign artificial monthly dates if needed for time-series analysis.The data is in matrix format: each column corresponds to a fund and each row to a monthly return observation. There are no separate factor returns included in `hfdata`; thus, by default, performance is evaluated in absolute terms (excess returns relative to zero). However, the functions allow specifying a matrix of factor returns if a risk-adjusted alpha is desired.

For example, we load the `hfdata` and examine its dimensions:

```
library(PeerPerformance)

#> Loading required package: parallel

#> Loading required package: sandwich

#> Loading required package: lmtest

#> Loading required package: zoo

#>
#> Attaching package: 'zoo'

#> The following objects are masked from 'package:base':
#>
#>     as.Date, as.Date.numeric

#> Loading required package: compiler

data("hfdata")
dim(hfdata)

#> [1]  60 100

hfdata[1:5, 1:6] # display first 5 rows and 6 columns

#>            Fund 1       Fund 2      Fund 3       Fund 4     Fund 5      Fund 6
#> [1,]  0.026288712  0.032836280 0.021296415 0.0258326502 0.01291151 0.022684554
#> [2,]  0.019421945  0.036557762 0.012542671 0.0270003125 0.02617951 0.016551571
#> [3,]  0.024603785  0.029981022 0.027115374 0.0182531033 0.02043003 0.011307394
#> [4,] -0.009895158 -0.010718460 0.004974697 0.0007281787 0.01754100 0.009935132
#> [5,]  0.023326055  0.004152093 0.020099802 0.0032256162 0.03195554 0.003578033
```

### 3.2 Peer Performance Analysis Functions

The main function for peer performance analysis is `alphaScreening()`. This function computes the alpha outperformance ratios for a set of funds, using the methodology described earlier (multiple pairwise tests and FDR adjustment). Despite the name "alpha" (historically because it compares intercepts), it can operate with or without explicit factor data:

• If a matrix of factor returns is provided via the factors argument, `alphaScreening` runs a regression for each pair of funds to estimate $\alpha_{i-j}$ as in the model above, using HAC (heteroskedasticity and autocorrelation consistent) standard errors by default for the intercept. This gives a risk-adjusted comparison controlling for the specified factors.

• If factors = NULL (the default), it compares raw returns, effectively treating the sample mean return as the performance metric for each fund (which is equivalent to using an intercept-only model).

The function returns a list containing, for each fund in the input, the estimated performance ratios and related statistics. Key components of the result include:

• **$alpha**: the estimated performance measure for each fund (if factors were provided, this is the fund's estimated alpha; if not, it may just be the average return or an NA if not applicable).

• **$pipos**, **$pizero**, **$pineg**: Numeric vectors of length $N$ giving the estimated $\hat{\pi}_i^+$, $\hat{\pi}_i^0$, and $\hat{\pi}_i^-$ for each fund $i$. These are the outperformance, equal-performance, and underperformance ratios (in proportion, between 0 and 1).

• **$pval**: an $N \times N$ matrix of pairwise p-values (or sometimes a compressed object) for the tests $H_0 : \Delta_{i,j} = 0$. This is provided mostly for transparency; the user typically doesn't need to inspect all these p-values.

• **$lambda**: the chosen $\lambda_i$ thresholds for each fund used to compute $\hat{\pi}_i^0$.

• **$n0**: the estimated number of equal performers $\hat{n}_i^0$ for each fund.

• Additional elements like **$z** or **$tstat** might contain test statistics.

The computation can be intensive for large $N$ because it involves $N(N-1)/2$ regressions or tests. Therefore, `alphaScreening` is parallelized: users can specify `control = list(nCore = k)` to use $k$ CPU cores for the pairwise computations. Other control options include `hac = TRUE/FALSE` to enable or disable the HAC robust standard error (by default, it is TRUE, using Newey-West adjustment for the regressions), and significance levels for Sharpe screening (not used in `alphaScreening`, but in `sharpeScreening`).

Here is a quick example of `alphaScreening` on a subset of the data (for brevity, we use 5 funds). We disable parallelization (nCore = 1) just for this small example:

```
# Peer performance screening on 5 funds (columns 15-20 of hfdata)
rets <- hfdata[, 15:20]
result <- alphaScreening(rets, control = list(nCore = 1))
# Examine the estimated performance ratios for each fund:
round(result$pizero, 3)  # equal-performance ratios

#> [1] 0.000 0.000 0.000 0.000 0.334 0.286

round(result$pipos, 3)   # outperformance ratios

#> [1] 0.600 0.000 0.400 0.200 0.666 0.714

round(result$pineg, 3)   # underperformance ratios

#> [1] 0.4 1.0 0.6 0.8 0.0 0.0
```

As expected, each fund's ratios add up to 100% of its peer group. The output indicates, for instance, that Fund1 has $\hat{\pi}_1^+ = 0.600$ and $\hat{\pi}_1^- = 0.400$, suggesting it outperforms 60% of the other 4 funds and underperforms 40% of them. In a real analysis, one might focus on funds with particularly high outperformance ratios and low underperformance ratios as the consistently superior funds.

If we had prior factors (say we know all these hedge funds are equity-oriented and we have market index returns), we could include them via factors argument to ensure we compare alphas on a level playing field. For example, if factor.mat held the corresponding market returns for 60 months, we could do alphaScreening(rets, factors=factor.mat, control=...). In this vignette, we proceed without factor adjustments for simplicity.

### 3.3 Sharpe Ratio Comparison Functions

The **PeerPerformance** package also provides functions to compute Sharpe and modified Sharpe ratios:

- `sharpe(R)` computes the Sharpe ratio for each column (fund) in return matrix R.
- There is also `msharpe(R, level = 0.95)` which computes the modified Sharpe ratio at a specified confidence level (e.g., 95% by default).

These functions return a numeric vector of length equal to the number of funds. For example:

```
sharpe_values <- sharpe(rets)
msharpe_values <- msharpe(rets, level = 0.95)
print(round(msharpe_values, 3))

#> Fund 15 Fund 16 Fund 17 Fund 18 Fund 19 Fund 20
#>   0.114  -0.067   0.046   0.007   0.209   0.488
```

This shows each fund's modified Sharpe ratio using 95% VaR. A higher value indicates better risk-adjusted performance accounting for tail risk.

There are also Sharpe ratios comparison functions in the **PeerPerformance** package:

• `sharpeTesting(x, y)` conducts a hypothesis test for $H_0 : SR_x = SR_y$ vs $H_1 : SR_x \neq SR_y$. By default, it may use a large-sample approximation; users can supply `control = list(bootstrap = TRUE, nBootstrap = 1000)` to perform a Ledoit and Wolf (2008) bootstrap with 1000 resamples for more accurate p-values (especially if the return series are not very long or are autocorrelated). The result is an object of class `htest` (similar to a t-test result) containing a p-value. A small p-value (below 0.05) indicates a significant difference in Sharpe ratios.

• `msharpeTesting(x, y, level=0.95)` tests equality of modified Sharpe ratios at confidence level *level*. It returns an `htest` object with a p-value for the null that the two modified Sharpe ratios are equal. By default, it uses the asymptotic method from Ardia and Boudt (2015). Users can also request a bootstrap via the control list (similar to above) if needed.

For example, let's compare Fund1 and Fund2 from our dataset in terms of modified Sharpe ratio (at 95% confidence):

```
# Test if Fund1 and Fund2 have equal modified Sharpe ratio
res_test <- msharpeTesting(hfdata[,1], hfdata[,2], level = 0.95)
res_test$pval
```

```
#> [1] 0.6445251
```

In this example, the $p - value$ is 0.64 > 0.05 so we reject the null hypothesis and conclude that Fund1 and Fund2 don't have statistically different modified Sharpe ratios.

There are also screening functions `sharpeScreening(R, ...)` and `msharpeScreening(R, ...)` which mirror alphaScreening but use Sharpe or modified Sharpe ratios for peer comparisons. For instance, `sharpeScreening(rets)` would give each fund's "Sharpe outperformance ratio" – the proportion of peers it has a higher Sharpe ratio than, adjusted for luck. methodologically, this uses bootstrap tests for each pair's Sharpe difference, which can be time-consuming, so parallel computing is supported via `control = list(nCore = ...)`. The usage and output structure (pipos, pizero, pineg) are analogous to the alpha case. In practice, one might use Sharpe screening when factor models are not available or when interested in total risk-adjusted returns, and use alpha screening when factor-adjusted alphas are preferred.

### 3.4 Additional Implementation Notes

The package functions are designed to be relatively easy to use for practitioners. In summary:

• Parallelization: The argument `control = list(nCore = k)` can be passed to any of the screening functions (`alphaScreening`, `sharpeScreening`, `msharpeScreening`) to run the pairwise tests on $k$ CPU cores in parallel, which is highly recommended for large universes of funds (since the number of tests grows quadratically with $N$). By default, `nCore` might use a single core, so be sure to increase it on a multi-core machine for speed.

• HAC vs Bootstrap: For alpha differences, the default uses HAC standard errors (Newey-West) for inference on $\alpha_{i-j}$. For Sharpe differences, a more brute-force bootstrap is used by default (due to the complexities of Sharpe ratio distribution). The user can choose asymptotic methods or bootstraps via the control parameters. The package's defaults generally follow the recommendations of the authors: e.g., use FDR for peer ratios, use block bootstrap for Sharpe tests, etc.

• Interpretation: The output of screening functions can be directly used to rank or filter funds. For instance, we might select funds with $\hat{\pi}_i^+ > 0.5$ and $\hat{\pi}_i^- < 0.1$ as candidates for top performers (meaning a fund beats over half its peers significantly and is beaten by very few). The package also suggests plotting capabilities: while no dedicated plotting function is provided, one can easily create a ternary plot or bar chart of the three ratios for each fund to visualize their peer performance profile. In their paper, Ardia and Boudt (2018) present "peer performance screening plots" where each average fund outperformance, underperformance, and equal-performance ratios are represented as shaded areas defined by $(\pi^0, \pi^+, \pi^-)$ coordinates. Such visualizations can be manually produced using the results.

• Limitations: The methodology assumes a relatively stable performance measure over the sample considered. If performance is highly time-varying, the concept of a single $\pi_i^+$ over a long period may be less meaningful (one might then consider rolling-window analyses). Also, the FDR method assumes that under the null of equal performance, p-values are independent or only weakly dependent across

peers; extreme dependence (e.g., one fund outperforming all others by the same large margin) can in theory affect the estimator's consistency, though this is a pathological case.

Having covered the implementation details, we now turn to a complete example to illustrate how one might use **PeerPerformance** in practice.

## 4 Example: Hedge Fund Peer Performance Analysis

In this section, we walk through a realistic (albeit simplified) example of using the **PeerPerformance** package to analyze a set of hedge funds. We use the provided `hfdata` dataset as our universe of funds. For demonstration purposes, suppose we treat these 100 funds as one peer group (perhaps they are all following a similar strategy, like "Equity Hedge" style). Our goals are:

1. Compute the equal-, out-, and underperformance ratios for each fund.
2. Identify a few funds that appear to outperform most of their peers.
3. Verify differences in performance with a Sharpe ratio test for an illustrative pair.

After loading the package and data (done earlier), we perform the peer performance screening using the default settings (no factor adjustments, HAC errors):

```
# Run alpha screening with parallelization
res <- alphaScreening(hfdata, control = list(nCore = 15))

# Extract performance classification ratios
pipos  <- res$pipos
pizero <- res$pizero
pineg  <- res$pineg

# Combine all ratios into a matrix with proper column names
ratios <- cbind(pipos, pizero, pineg)
colnames(ratios) <- c("pipos", "pizero", "pineg")

# Add fund numbers as a column
fund_id <- 1:nrow(ratios)
ratios_with_id <- cbind(Fund = fund_id, ratios)

# Sort by outperformance and display top 10
ord <- order(pipos, decreasing = TRUE)
head(ratios_with_id[ord, ], 10)

#>       Fund     pipos    pizero pineg
#>  [1,]   35 0.8232323 0.1767677     0
#>  [2,]   99 0.8181818 0.1818182     0
#>  [3,]   50 0.7979798 0.2020202     0
#>  [4,]   51 0.7777778 0.2222222     0
#>  [5,]   49 0.7727273 0.2272727     0
#>  [6,]   73 0.7306397 0.2693603     0
#>  [7,]   12 0.6632994 0.3367006     0
#>  [8,]   23 0.5959335 0.4040665     0
#>  [9,]   41 0.4949495 0.5050505     0
#> [10,]   29 0.4191909 0.5808091     0
```

The results show that among the 100 funds in our universe, Fund35 dominates approximately 82% of them, and is statistically worse than only 18%. Thus, if we intend to pick the top performer in relative terms among our universe, we should select Fund35.

As a cross-check, we might want to confirm that Fund35 indeed has a significantly higher Sharpe ratio than a bottom ranked-fund. We can use `sharpeTesting` for a direct Sharpe ratio comparison:

```
# Inspect first 10 funds
print(ratios_with_id[1:10, ])

#>       Fund       pipos     pizero      pineg
```

```
#>  [1,]    1 0.000000e+00 0.70709057 0.29290943
#>  [2,]    2 0.000000e+00 0.73267999 0.26732001
#>  [3,]    3 2.775558e-17 0.92870764 0.07129236
#>  [4,]    4 2.775558e-17 0.83792017 0.16207983
#>  [5,]    5 0.000000e+00 1.00000000 0.00000000
#>  [6,]    6 0.000000e+00 0.60606061 0.39393939
#>  [7,]    7 1.010101e-02 0.02525253 0.96464646
#>  [8,]    8 0.000000e+00 0.23569024 0.76430976
#>  [9,]    9 0.000000e+00 0.62626340 0.37373660
#> [10,]   10 0.000000e+00 0.69023969 0.30976031
```

```
# Select Fund7 for example
x <- hfdata[, 35]
y <- hfdata[, 7]

# Apply sharpeTesting
test_res <- sharpeTesting(x, y, control = list(bootstrap = TRUE, nBootstrap = 1000))
test_res$pval
```

```
#> [1] 0.004060051
```

The $p - value$ of 0.004 strongly rejects equal Sharpe ratios, indicating Fund35's Sharpe ratio is significantly higher than Fund7's. This aligns with our screening conclusion that Fund35 is far superior. In practice, one would perform such pairwise tests sparingly or as needed (since we already have a multiple comparisons procedure in the screening, doing many separate tests is not usually necessary unless for presentation or additional verification).

Finally, we demonstrate the modified Sharpe ratio test on the same pair of funds. Modified Sharpe is useful if returns are not normally distributed. In our case, the Sharpe Ratio test already differentiated the two funds, so the Modified Sharpe test serves as an additional validation that we have a significant difference between the two funds. In the case where the two funds had a moderate difference, this test would permit to see if perhaps normal Sharpe wouldn't flag them but modified Sharpe would.

```
mtest_res <- msharpeTesting(x, y, level = 0.95)
mtest_res$pval
```

```
#> [1] 0.006405725
```

The $p - value$ is very low again but it is different than the one from the normal Sharpe Ratio test, indicating that $x$ and $y$ might have non-normal returns, but still have a significant difference under these conditions. In another case, perhaps the normal Sharpe test would have had $p - value = 0.08$ (not significant at 5%), but the modified Sharpe test would have given $p - value = 0.03$, detecting a difference due to heavier tails in one fund's returns. This kind of discrepancy (as noted by Ardia and Boudt (2015)) underscores the importance of using the right performance measure for the data at hand.

## 5 Conclusion

The **PeerPerformance** package provides an academically rigorous toolkit for performance analysis among a set of investment funds. By focusing on peer performance ratios, it moves beyond simplistic rankings and addresses the prevalence of "equitably performing" funds through a sophisticated multiple testing approach. The equal-performance, outperformance, and underperformance ratios offer a more nuanced view of how a fund stands relative to its peers, with built-in control for false discoveries. We demonstrated how to compute these ratios in R and interpret them to identify truly skilled funds versus lucky ones. We also showed how the package can test Sharpe and modified Sharpe ratios to complement the analysis with classical risk-adjusted metrics.

In practice, analysts can use **PeerPerformance** to screen a broad universe of funds, flagging those that consistently outperform a large portion of their peers, while avoiding being misled by funds that merely benefited from luck. The statistical rigor (e.g., using HAC standard errors, bootstrap, FDR) is largely under-the-hood; the user interacts with simple R functions and interprets intuitive outputs (percentages of peers). This lowers the barrier to applying advanced performance analytics in the financial industry.

In summary, **PeerPerformance** implements the latest research in luck-corrected performance evaluation (Ardia and Boudt, 2018) and provides practical tools for multiple comparisons in performance analysis, enabling more reliable identification of truly outperforming funds.

## References

D. Ardia and K. Boudt. Testing equality of modified sharpe ratios. *Finance Research Letters*, 13:97–104, 2015. doi: 10.1016/j.frl.2015.02.008. [p1, 4, 7, 9]

D. Ardia and K. Boudt. The peer performance ratios of hedge funds. *Journal of Banking & Finance*, 87: 341–353, 2018. doi: 10.1016/j.jbankfin.2017.11.007. [p1, 2, 4, 7, 9]

L. Favre and J.-A. Galeano. Mean-modified value-at-risk optimization with hedge funds. Working paper, University of Lausanne and UBS Warburg, 2002. Available at SSRN: https://ssrn.com/abstract=371202. [p1, 4]

G. N. Gregoriou and J.-P. Gueyie. Risk-adjusted performance of funds of hedge funds using a modified sharpe ratio. *Journal of Wealth Management*, 6(3):77–83, 2003. doi: 10.3905/jwm.2003.442312. [p1, 4]

O. Ledoit and M. Wolf. Robust performance hypothesis testing with the sharpe ratio. *Journal of Empirical Finance*, 15(5):850–859, 2008. doi: 10.1016/j.jempfin.2007.12.002. [p1, 4, 7]

J. D. Storey. A direct approach to false discovery rates. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(3):479–498, 2002. doi: 10.1111/1467-9868.00346. [p2]

*David Ardia*
*CIRANO, GERAD, HEC Montreal*
*Department of Decision Science*
*Montreal, Canada*
david.ardia.ch@gmail.com

*Benjamin Seguin*
*HEC Montreal*
*Department of Decision Science*
*Montreal, Canada*
benjamin.seguin@outlook.fr