PEER REVIEW RECORD

Taking R to its limits: 90+ tips

Michail Tsagris and Manos Papadakis

Vol. 1 (2026), Art. 15 · https://doi.org/10.68365/jrepstat.2026.15

The decision letters, the reports the authors read, their answers and the editor's signed decisions, as written. The reports come from the journal's review engine. The authors could have kept this record private until they confirmed the proof.

DECISION PATH

2026-10-02	Major revision
2026-10-22	10 answers
2026-10-22	Minor revision
2026-11-09	16 answers
2026-11-14	Accepted

Round 1: Major revision

SIGNED DECISION

Confirmed and signed by Matthias Templ on 2026-10-02 · ORCID 0000-0002-8638-5276

DECISION LETTER

Thank you for your submission to JRepStat. We are inviting a major revision. The reviewers' points follow, grouped by weight.

A tutorial that compiles more than ninety practical tips for writing faster, lower-memory, and numerically stable R code, aimed at statistical-computing practitioners who run simulations and matrix-heavy analyses and want speed without rewriting in C++. We would be glad to see the revised version.

How to read this letter. The decision rests on the items marked required. If a required item seems wrong to you, or cannot be done within this paper, say so in your response letter and give the reason. A reasoned decline is a legitimate answer, and the editor weighs it. Reviewers can be mistaken. Items marked recommended or optional are advice from the reviewers. Adopt them, decline them with a sentence, or leave them for later work. None of that counts against the paper. The reviewer reports below carry the same marks. The editor who signs this decision may weigh an item differently in the accompanying message, and that message prevails.

REQUIRED (the decision rests on these):

- [consistency-check] The abstract and the introduction both state "more than 100 tips", but the body contains 94 numbered tips (1 through 94, the last being the [[versus $ item in Section 11). The title says "90+ tips", which matches the count. Reconcile the abstract and the two introduction sentences ("a collection of more than 100 tips", "We have listed more than 100 tips") with the actual number so the claim agrees with the title and the content.
- [methods-soundness] Tip 64 states that prcomp() "performs the eigendecomposition of a matrix". prcomp() computes principal components by singular value decomposition of the (centered) data matrix, while princomp() uses the eigendecomposition of the covariance matrix. Please correct the description so the recommendation and its justification match what the functions do.
- [Reviewer 1] State the BLAS in use and add a reproducibility script for the reported timings. Several matrix comparisons (crossprod vs t(X) %*% Y, princomp vs prcomp, float throughput) depend on which BLAS is linked, and reference BLAS results can invert under OpenBLAS or MKL. Provide the benchmark script plus sessionInfo and the BLAS identity, or qualify these comparisons as reference-BLAS results.

- [Reviewer 2] Add a competing-interest statement. The authors maintain Rfast, Rfast2 and estats, which the paper recommends as faster (tips 64, 72, Section 10).
- [Reviewer 2] Where an authored package is recommended as faster, compare it against the strongest alternative, not base R alone. Either benchmark Rfast/estats functions against matrixStats, collapse and data.table, or soften the superiority claims to "an alternative to base R".
- [Reviewer 2] Ship a runnable benchmark script that regenerates the two printed tables (tips 16 and 46) with a fixed seed, the R version, and the BLAS named, or remove the printed numbers. The uploaded file reproduces code chunks but does not run end to end.
- [Reviewer 2] Correct tip 84: values below .Machine$double.eps (2.22e-16) are not rounded to zero. Underflow to zero happens near 1e-308 (and subnormals to 5e-324). The log.p advice stands, the stated threshold does not.
- [Reviewer 3] Tip 17 option a3 divides `rowsum(x, ina, reorder = FALSE)` by `as.vector(table(ina))`, whose counts are in sorted group order, so the two are misaligned when groups do not first appear in sorted order. Fix the alignment or remove a3.
- [Reviewer 3] Add data.table coverage beyond `fread`: keyed joins, grouped aggregation with `[, by = ]`, and update by reference, cross-referenced from tips 12 and 17.
- [Reviewer 3] A short note on when to leave base R for Rcpp or RcppArmadillo yourself, not only via a wrapper package.

RECOMMENDED:

- [methods-soundness] Most matrix timing claims (for example crossprod versus t(X) %*% Y in Tips 52 to 53, princomp versus prcomp in Tip 46, and the float and Matrix tips) depend on the BLAS/LAPACK backend and the platform. Under an optimized multithreaded BLAS several of these gaps shrink or reverse. One sentence stating that the benchmarks were run with the reference BLAS on the described laptop, and that margins vary with the linear algebra backend, would set the reader's expectations correctly.
- [methods-soundness] Tip 64 says at most n eigenvalues of the correlation matrix are non-zero when $n < p$. After centering and standardizing, the rank is at most n - 1, so at most n - 1 eigenvalues are non-zero. Adjust the statement, since the surrounding argument about hd.eigen() rests on this rank.
- [methods-soundness] Several timing displays (Tips 16 and 46, the profiling output in Tip 2) report microbenchmark or Rprof results, but the data-generating lines and any seed are given only for some. Adding the generating code and a set.seed for each reported benchmark would let a reader reproduce the numbers.
- [statistical-rigor] Several quantitative speed claims carry no measurement: "at least 10 times faster" for Newton-Raphson versus nlm (tip 77), "3-4 times faster" for the G2 tests (tip 15), "more than 2 times faster" for sum(x %*% m) (tip 54), "almost 2 times faster" for dim over nrow (tip 42), and the many "much faster" statements in Section 10. Either give the timing that supports each figure or state the number as an approximate observation on the authors' machine.
- [statistical-rigor] The two benchmark tables (tips 16 and 46) report min, lower quartile, mean, median, upper quartile and max over 20 replications but no reasoning for 20 runs and no statement of the run-to-run spread of the difference. For prcomp versus princomp the maxima (50.55 and 51.37 ms) nearly coincide while the medians differ by about 18 ms, so a reader cannot tell how stable the ordering is. Report the paired difference and its spread, and note that a single laptop with JIT and BLAS settings fixed limits how far the ratios generalise.
- [writing-quality] In the introduction, "Gillespie and Lovelace (2016) gave a book–length treatment in his book" refers to two authors, so "his" should read "their". The sentence also repeats "book" three times (Wickham "in his book", Burns "in his book The R Inferno", "a book–length treatment in his book"); one pass to vary the wording would read more cleanly.
- [writing-quality] Tip 80 contains a spelling error, "afraind", that reads as an author typo rather than an extraction artefact. Please correct to "afraid".
- [Reviewer 1] Quantify a representative set of the bare "faster" claims (for example tips 42, 54, 89) with the same benchmark script, so the factors ("almost 2 times", "more than 2 times faster", "10-50 times") are supported.
- [Reviewer 2] Add collapse (fmean, fsum, fvar) to the grouped-operation tips (17, 38).

- [Reviewer 3] Note the numerical-stability cost of `princomp` over `prcomp` in tip 46, consistent with the paper's own stability section.
- [Reviewer 3] Add a note on reproducible parallel RNG (`clusterSetRNGStream`, `mc.set.seed`) in Section 9, given the simulation use case the paper motivates.
- [Reviewer 3] State the counter-case for the micro-optimizations that currently read as absolute (tip 42 `dim` vs `nrow`, tip 12 `mean` accuracy).

OPTIONAL:

- [methods-soundness] Tip 46 recommends princomp over prcomp on speed grounds, while the numerical-overflow section stresses stability. princomp via the covariance eigendecomposition is less accurate for ill-conditioned data than the SVD used by prcomp. A short note on when the speed gain is safe would align the two parts of the paper.
- [statistical-rigor] A comparison across more than one machine, operating system and BLAS backend would show which of the reported ratios are properties of R and which are properties of this particular laptop.

We will correct these in production, and you confirm them in the proof:

- [consistency-check] The profiling output in Tip 2 lists a function `"h"` (self.time 0.01) that does not appear in the function `f` shown just above it. The self-times still sum correctly (8.56 + 0.84 + 0.01 = 9.41), so the total is internally consistent, but the `"h"` row has no source in the displayed code. Confirm the intended output.
- [writing-quality] Section 10 has a dropped package name: "The same is true for ordinal logistic regression in and in ordinal (Christensen, 2015)". A package name is missing before the first "in" (the comparison names only one of the two packages being compared). Please supply the intended package.
- [Reviewer 1] The R Inferno (Patrick Burns) is cited in the introduction but has no entry in the references. Supply the reference.

REVIEWS

Report Reviewer 1, recommends minor

Written by the journal's review engine in the role of the application reviewer, and read by the editor before the decision. How the engine works is on the peer review page.

This is a useful tutorial. The problem is real: R users who run simulations or matrix-heavy analyses routinely leave large speed and memory gains on the table, and a single well-organised list of concrete idioms has clear value for that audience. The coverage is broad and mostly accurate. The numerical-overflow section (tips 79 to 88) is the strongest part. The inverse-logit reformulation, the softplus formula max(x,0) + log1p(exp(-abs(x))), the log-sum-exp trick, working in log-densities, and using log.p for sortable tail probabilities are all correct and are exactly the things practitioners get wrong. The matrix-algebra section is also solid: crossprod/tcrossprod, solve(X, Y) over solve(X) %*% Y, backsolve/forwardsolve for triangular factors, and the associativity point in tip 61 are all sound and well chosen.

My main concern as an application reviewer is that many speed claims are asserted without evidence, and the few that carry benchmarks were run on a single laptop with an unstated BLAS. This matters because a good share of the matrix tips are BLAS-bound. Whether crossprod(X, Y) beats t(X) %*% Y, and by how much, and whether princomp beats prcomp, depends on whether R is linked against reference BLAS, OpenBLAS, or MKL. On an optimised multi-threaded BLAS some of these gaps shrink or reverse. Readers will apply these tips on machines unlike yours, so the paper should say which BLAS produced the numbers and should ship the script that produced tables 16 and 46. That also lets the quantified factors scattered through the text ("almost 2 times faster", "more than 2 times faster", "10-50 times faster") be checked. A short reproducibility script with sessionInfo would settle all of this at once and is the natural minimum for a tutorial whose value is its timings.

One consistency point deserves attention because the paper itself raises the standard. Section 7 states that one should not sacrifice numerical stability for speed. Tip 46 then recommends princomp over prcomp on speed grounds alone. princomp works from the covariance matrix and its eigendecomposition, which

loses precision relative to the SVD of the centered data that prcomp uses, and the difference shows up on ill-conditioned or nearly collinear data that applied users meet often. A single sentence flagging the tradeoff keeps the tip while honouring the paper's own principle. A smaller correctness note: tip 36 gives x * (x > 0) for ifelse(x > 0, x, 0), which differs in the presence of NA, and tip 87′s integer point is right and could add that length/dim/nrow return integers so n * n overflows silently, which the example already shows.

The headline count needs fixing. The abstract and introduction say "more than 100" tips, the title says "90+", and the list runs to 94. The claim that this is the first compilation of such length depends on getting that number straight, so please set it to what the paper actually contains. The R Inferno is discussed in the introduction but is absent from the references and should be added. On presentation, the tips read as a flat sequence within each section, and a few (38 on apply for medians, 48 on by) qualify their own advice in ways that could confuse a reader skimming for a rule. A one-line summary table grouping the tips by theme would help the tutorial's usability, though the paper stands without it.

Numbered suggestions to the authors:

1. REQUIRED Set the tip count consistently across the title, abstract, and introduction to the number actually present (94 by my count), since the "first such list" claim rests on it.
2. REQUIRED State the BLAS used for the benchmarks and provide the script plus sessionInfo that produced the timing tables (tips 16 and 46). Note that crossprod vs explicit transpose-multiply, princomp vs prcomp, and the float throughput claim are BLAS-dependent and can narrow or invert under OpenBLAS or MKL, so present them as reference-BLAS results or qualify them.
3. REQUIRED Add one sentence to tip 46 noting that princomp is less numerically stable than prcomp because it eigendecomposes the covariance matrix, so the speed gain trades against precision on ill-conditioned data.
4. [editor_fix] Add a references entry for The R Inferno (P. Burns), which is cited in the introduction but missing from the reference list.
5. RECOMMENDED Support the unquantified speed factors for a representative subset (for example tips 42, 54, and 89) using the same benchmark script, so the stated multipliers are backed by numbers.
6. RECOMMENDED Note in tip 36 that x * (x > 0) and ifelse(x > 0, x, 0) differ when x contains NA, so readers do not swap them blindly.
7. OPTIONAL A short table grouping the tips by theme (memory, vectorization, matrix algebra, numerical stability) would make the tutorial easier to navigate and reference.

Report Reviewer 2, recommends major

Written by the journal's review engine in the role of the methodology reviewer, and read by the editor before the decision. How the engine works is on the peer review page.

This is a tutorial, and I review it as one. A curated list of idioms does not owe a benchmark for every line the way a methods paper owes one for its headline result. The numerical-overflow section (tips 79 to 88) is the strongest part: the inverse-logit reformulation, the softplus identity max(x,0)+log1p(exp(-|x|)), the log-sum-exp shift, log1p and expm1, and working with log-densities are all correct and well chosen. The vectorization and preallocation advice is sound and the worked profiling example makes its point cleanly.

My concerns are methodological and they cluster. First, the paper is written by the maintainers of Rfast, Rfast2 and estats, and it recommends those packages as faster (tips 64 and 72, and Section 10). There is no competing-interest statement, and that needs to be added. Tied to it is a fairness question that runs through the paper. Many tips present a fast option against the slowest baseline only. Where the recommended option is an authored package, the relevant comparison is against the strongest alternative in the ecosystem, which for column medians, variances and distances means matrixStats, collapse and data.table, and for a truncated eigendecomposition (tip 64) means something like RSpectra or irlba. Please either benchmark against those or phrase the recommendation as an alternative to base R. I have marked the comparison required with the softening as the low-cost route.

Second, reproducibility of the numbers you print. The paper shows two benchmark tables (centering in tip 16, princomp against prcomp in tip 46) and asserts ratios elsewhere ("more than 2 times faster",

"almost 2 times faster"). None of the matrix timings can be read without the BLAS, since reference BLAS, OpenBLAS and MKL change the relative cost of crossprod, solve, eigen and the float tip (70) by large factors. The environment gives the R version but not the BLAS, and the code file cannot regenerate the two tables because several chunks depend on objects defined only in the prose. A single script that builds the inputs, sets a seed, runs microbenchmark or bench, and prints the two tables would close this, or the printed numbers could be removed and the claims kept qualitative.

Third, two technical corrections. Tip 84 states that p-values below .Machine$double.eps are rounded to zero. That is off by roughly 290 orders of magnitude. A p-value of 1e-100 is representable and sortable. Underflow to zero happens near the smallest normal double, about 1e-308, with subnormals below that. The advice to carry log p-values is good, so only the justification needs fixing. Tip 46 recommends princomp over prcomp for speed. princomp forms and diagonalises the covariance matrix, which is the less stable route, while prcomp uses the SVD of the centered data. In a paper whose own motto is not to trade stability for speed, that recommendation needs one sentence of caveat. In the same spirit, tip 12 recommends sum(x)/length(x) over mean for speed, and mean's extra cost buys the two-pass accuracy you are giving up, which is worth a half-sentence.

A lighter point on the many micro-claims. "nrow is almost 2 times faster than dim(X)[1]" (tip 42), the internal .colSums (tip 39), iterating over elements (tip 93): these are true as ratios and negligible as absolute times. A reader can spend real effort chasing nanoseconds. Where you give a ratio without a measurement, either measure it or mark it as a constant-factor nicety.

Numbered suggestions to the authors:

1. REQUIRED Add a competing-interest statement naming Rfast, Rfast2 and estats as packages the authors maintain, since the paper recommends them (tips 64, 72, Section 10).
2. REQUIRED For each authored-package recommendation, compare against the strongest alternative (matrixStats, collapse, data.table, and RSpectra or irlba for tip 64), or reword the claim to "an alternative to base R" where you do not benchmark.
3. REQUIRED Supply a runnable script that regenerates the tables in tips 16 and 46 with a fixed seed, the R version, and the BLAS stated, or remove the printed timings and keep the claims qualitative.
4. REQUIRED In tip 84, replace the .Machine$double.eps threshold with the correct underflow behaviour near 1e-308 and the subnormal range, keeping the log.p recommendation.
5. REQUIRED Add one sentence to tip 46 that princomp diagonalises the covariance matrix and is less stable than the SVD-based prcomp, so the speed gain carries an accuracy cost.
6. [editor_fix] Make the tip count consistent: the abstract says "more than 100", the title says "90+", and the list ends at 94.
7. RECOMMENDED Add collapse (fmean, fsum, fvar) to the grouped-operation tips 17 and 38, where it is typically faster than rowsum and aggregate.
8. RECOMMENDED Temper the unquantified ratio claims (tips 42, 54, 70, 93) to qualitative statements, or back each with a measurement, and add a half-sentence to tip 12 on the accuracy that sum/length gives up relative to mean.
9. OPTIONAL A table of the matrix-tip speedups under reference BLAS, OpenBLAS and MKL would settle how much each recommendation depends on the linear algebra backend.

Report Reviewer 3, recommends minor

Written by the journal's review engine in the role of the software reviewer, and read by the editor before the decision. How the engine works is on the peer review page.

I read this as a software and reproducibility reviewer, with the uploaded R file treated as part of the submission. The tutorial is useful and most of the advice is correct and well chosen. The matrix-algebra tips (crossprod, solve(X, Y), backsolve, the Mahalanobis route to the multivariate normal exponent) and the numerical-overflow section (log-sum-exp, log1p, log-densities, log p-values) are the strongest parts and are the kind of thing many R users genuinely do not know. My concerns are about whether the shipped artifact does what the submission record says it does, one correctness bug, and a disclosure that a computing venue expects.

The declared entry point is `Rscript R code.R`, which signals a runnable artifact. It is not one. The file's own header states that several fragments refer to objects defined only in surrounding prose, and a top-to-

bottom run fails early: chunk 2 calls `microbenchmark(mean(x), ...)` with no `x` defined, chunk 5 uses `n`, chunk 6 uses `fast` and `slow`, and so on. No `set.seed` appears anywhere, so even the self-contained chunks that build data with `rnorm` do not reproduce fixed numbers. The two timing tables in the paper (centering in tip 16, princomp against prcomp in tip 46) are pasted into the code as comments and are not regenerated by any runnable, seeded code. The machine is named, which is good provenance, but timings also depend on the BLAS, the OS and the R build, none of which is pinned beyond the prose. For a paper whose entire value rests on speed claims, a reader should be able to run something and see the numbers appear. The cheapest honest fix is to split the file: keep the illustrative snippets as clearly non-executable, and add two or three small seeded scripts that build the inputs and print the benchmark tables. If the file is meant to be snippets only, the declared entry point should say so.

Tip 17 has a correctness bug in its "a bit faster" variant. `rowsum(x, ina, reorder = FALSE)` returns group rows in order of first appearance, while `as.vector(table(ina))` returns counts in sorted factor-level order. When the two orders differ, each group mean is divided by another group's count. The default `rowsum(x, ina)` in a2 is fine because it sorts. Please either align the counts to the `reorder = FALSE` order or drop a3. This belongs in a tutorial precisely because a reader will copy it.

The authors maintain Rfast, Rfast2 and estats, and the paper recommends these packages in tips 64, 72 and 74 and across Sections 9 and 10. A reader is entitled to know that. Please add a short interest declaration. This is not a judgement on the recommendations, several of which are reasonable, only on disclosing who wrote the packages being recommended.

Two smaller content points are worth lining up with the paper's own stated priorities. Section 7 argues that numerical stability should not be sacrificed for speed, yet tip 46 recommends `princomp` over `prcomp` purely on timing. `princomp` forms the covariance and eigendecomposes it, which is less accurate for near-collinear data than the SVD route in `prcomp`, and one sentence should say so. On data.table, the paper uses it only through `fread`. For the aggregation tips (12 and 17) and for large grouped work generally, `[, by = ]`, keyed joins and update by reference are often the fastest route available, and a tutorial on efficient R aimed at large data reads as incomplete without them. The tip count in the abstract and introduction ("more than 100") does not match the 94 numbered tips or the "90+" in the title.

Numbered suggestions to the authors:

1. REQUIRED Make `R code.R` run end to end with inputs defined and `set.seed` set, or re-declare it as non-executable snippets and add separate seeded scripts that regenerate the centering (tip 16) and princomp/prcomp (tip 46) tables.
2. REQUIRED Fix tip 17: align `rowsum(x, ina, reorder = FALSE)` with counts in the same order, or remove the a3 variant, because dividing by `as.vector(table(ina))` uses sorted-order counts.
3. REQUIRED Add a declaration of the authors' interest in Rfast, Rfast2 and estats where those packages are recommended.
4. [editor_fix] Reconcile the tip count: abstract and introduction say "more than 100", the list ends at 94, and the title says "90+".
5. RECOMMENDED Add data.table beyond `fread`: keyed joins, grouped aggregation with `[, by = ]`, and update by reference, linked from tips 12 and 17.
6. RECOMMENDED In tip 46, state that `princomp` is less accurate than `prcomp` for near-collinear data.
7. RECOMMENDED In Section 9, add a note on reproducible parallel RNG using `clusterSetRNGStream` and `mc.set.seed`.
8. RECOMMENDED Give the counter-case for the micro-optimizations that read as absolute, including tip 42 (`dim` vs `nrow`) and the accuracy cost in tip 12 (`mean` vs `sum(x)/length(x)`).
9. OPTIONAL Add a short note on when to move to Rcpp or RcppArmadillo yourself, as a complement to the wrapper-package route.

REVISION, RECEIVED 2026-10-22

Authors confirm each required point as done or give their reason for declining it. Further answers are voluntary, because the review compares every revision with the previous version and the next round's reports say what remains open.

We thank the reviewers for their time. We have addressed most of their comments. We further proof read the paper and corrected more typos, re-organized some parts, and deleted a few tips, we are now counting

91 tips. There is a comment though that we must ask. The reviewers ask us to write add a competing-interest statement since we are he maintainers of Rfast, Rfast2 and estats R packages and we mention these in the paper. Do we have to add this statement? If yes, can you please help us write it?

> REQUIRED The abstract and the introduction both state “more than 100 tips”, but the body contains 94 numbered tips (1 through 94, the last being the `[[` versus `$` item in Section 11). The title says “90+ tips”, which matches the count. Reconcile the abstract and the two introduction sentences (“a collection of more than 100 tips”, “We have listed more than 100 tips”) with the actual number so the claim agrees with the title and the content.
>
> **Done.**

> REQUIRED Tip 64 states that prcomp() “performs the eigendecomposition of a matrix”. prcomp() computes principal components by singular value decomposition of the (centered) data matrix, while princomp() uses the eigendecomposition of the covariance matrix. Please correct the description so the recommendation and its justification match what the functions do.
>
> **Done.**

> REQUIRED State the BLAS in use and add a reproducibility script for the reported timings. Several matrix comparisons (crossprod vs t(X) %*% Y, princomp vs prcomp, float throughput) depend on which BLAS is linked, and reference BLAS results can invert under OpenBLAS or MKL. Provide the benchmark script plus sessionInfo and the BLAS identity, or qualify these comparisons as reference-BLAS results.
>
> **Done.**

> REQUIRED Add a competing-interest statement. The authors maintain Rfast, Rfast2 and estats, which the paper recommends as faster (tips 64, 72, Section 10).
>
> **Done.**

> REQUIRED Where an authored package is recommended as faster, compare it against the strongest alternative, not base R alone. Either benchmark Rfast/estats functions against matrixStats, collapse and data.table, or soften the superiority claims to “an alternative to base R”.
>
> **Done.**

> REQUIRED Ship a runnable benchmark script that regenerates the two printed tables (tips 16 and 46) with a fixed seed, the R version, and the BLAS named, or remove the printed numbers. The uploaded file reproduces code chunks but does not run end to end.
>
> **Done.**

> REQUIRED Correct tip 84: values below .Machine$double.eps (2.22e-16) are not rounded to zero. Underflow to zero happens near 1e-308 (and subnormals to 5e-324). The log.p advice stands, the stated threshold does not.
>
> **Done.**

> REQUIRED Tip 17 option a3 divides `rowsum(x, ina, reorder = FALSE)` by `as.vector(table(ina))`, whose counts are in sorted group order, so the two are misaligned when groups do not first appear in sorted order. Fix the alignment or remove a3.
>
> **Done.**

> REQUIRED Add data.table coverage beyond `fread`: keyed joins, grouped aggregation with `[, by = ]`, and update by reference, cross-referenced from tips 12 and 17.
>
> **Done.**

> REQUIRED A short note on when to leave base R for Rcpp or RcppArmadillo yourself, not only via a wrapper package.
>
> **Done.**

The authors’ full response letter, as uploaded, follows as Appendix A.

Round 2: Minor revision

SIGNED DECISION

Confirmed and signed by Matthias Templ on 2026-10-22 · ORCID 0000-0002-8638-5276

Confirm of the draft. Eight required points remain, four of them carried over from round 1. The editor intends this to be the last round and accepts the lighter routes the letter offers for the benchmark points.

DECISION LETTER

Thank you for the revised version of your manuscript. This is round 2, and we are inviting a further minor revision. Because you were asked for a major revision, this round read the new version whole, and the editor has read your response against it. What follows is our reading of this version. An item carried over from the previous round says so.

A tutorial compiling 91 practical tips for writing faster, lower-memory, and numerically stable R code, aimed at statistical-computing practitioners who run simulations and matrix-heavy analyses and want speed without rewriting in C++. We would be glad to see the revised version.

How to read this letter. The decision rests on the items marked required. If a required item seems wrong to you, or cannot be done within this paper, decline it on the revision page and give the reason. A reasoned decline is a legitimate answer, and the editor weighs it. Reviewers can be mistaken. Items marked recommended or optional are advice from the reviewers. Adopt them, decline them with a sentence, or leave them for later work. None of that counts against the paper. The reviewer reports below carry the same marks. The editor who signs this decision may weigh an item differently in the accompanying message, and that message prevails.

REQUIRED (the decision rests on these):

- [novelty-and-prior-art] The nearest prior work is the authors' own earlier preprint "Taking R to its limits: 70+ tips" (Tsagris and Papadakis, 2018, PeerJ Preprints, doi:10.7287/peerj.preprints.26605v1), which the search found and which does not appear in the reference list. The current manuscript states "to the best of our knowledge, this is the first time such a long list has been compiled in one place," and the response letter itself refers to "an earlier version of this paper, which can be found on the web." That earlier version already compiled a comparable list, so the first-compilation claim is not accurate as written without citing and positioning against it. Please cite the 2018 preprint and state what the present version adds relative to it (the count moving from 70+ to 91, the numerical-stability section, the data.table and matrix additions), or qualify the first-compilation sentence.
- [theory-and-proofs] Tip 17 option a3 still misaligns in the same family of cases the prior round flagged. The author removed `reorder = FALSE`, so `rowsum(x, ina)` now returns groups in sorted order and divides by `as.vector(tabulate(ina))`. This agrees only when the group codes are the consecutive positive integers 1..k. When the codes have a gap, say `ina = c(1,3,3,1)`, `rowsum` returns two rows while `tabulate(ina)` returns `c(2,0,2)` of length three, and the division recycles, so each group mean is divided by the wrong count. Option a2 with `table(ina)` is safe because `table` returns only present levels in sorted order. Either drop a3, or add the "consecutive strictly positive integers" caveat that Tip 16 already states for `tabulate` and cross-reference it.
- [ethics-and-integrity] The manuscript still carries no competing-interest statement. The authors maintain Rfast, Rfast2 and estats, and the paper recommends these packages as faster in tips 62 (hd.eigen), 70 (Rfast::Dist), 72 (colMedians) and across Sections 9 and 10. Three reviewers asked for this in the prior round, and the response letter defers it ("We will ask the editors what to write and where") rather than adding it. The disclosure is required regardless of how the recommendations are judged. Add a short statement naming Rfast, Rfast2 and estats as packages the authors maintain, placed before the references. A usable form: "M. Tsagris and M. Papadakis are the maintainers of the R packages Rfast, Rfast2 and estats, which are recommended in this paper. The authors declare no other competing interests."
- [Reviewer 2] Add the competing-interest statement. It is still absent from the manuscript, and the paper recommends Rfast, Rfast2, estats and hd.eigen. A one-line declaration suffices and I give wording below.
- [Reviewer 1] State the BLAS in use and add a reproducibility script for the reported timings. Several matrix comparisons (crossprod vs t(X) %*% Y, princomp vs prcomp, float throughput) depend on which BLAS is linked, and reference BLAS results can invert under OpenBLAS or MKL. Provide the

benchmark script plus sessionInfo and the BLAS identity, or qualify these comparisons as reference-BLAS results. This was asked in the previous round and is still open.
- [Reviewer 2] Where an authored package is recommended as faster, compare it against the strongest alternative, not base R alone. Either benchmark Rfast/estats functions against matrixStats, collapse and data.table, or soften the superiority claims to "an alternative to base R". This was asked in the previous round and is still open.
- [Reviewer 2] Ship a runnable benchmark script that regenerates the two printed tables (tips 16 and 46) with a fixed seed, the R version, and the BLAS named, or remove the printed numbers. The uploaded file reproduces code chunks but does not run end to end. This was asked in the previous round and is still open.
- [Reviewer 3] Add data.table coverage beyond `fread`: keyed joins, grouped aggregation with `[, by = ]`, and update by reference, cross-referenced from tips 12 and 17. This was asked in the previous round and is still open.

RECOMMENDED:

- [methods-soundness] The matrix-timing recommendations (crossprod versus t(X) %*% Y in tips 50 to 51, princomp versus prcomp in tip 44, the float tip 68) are presented as general orderings, and the authors' own response notes they have previously seen prcomp run faster than princomp on other hardware. The revision adds the LAPACK version (3.12.1) in the introduction but not the BLAS identity, nor the one sentence that these margins depend on the linear algebra backend and can narrow or reverse under an optimized multithreaded BLAS. Adding that sentence would make the printed orderings match what they are, observations under one backend. This partially addresses d78ce440.
- [statistical-rigor] The paper's quantitative speed multipliers are still stated as bare factors with no measurement behind them: "3-4 times faster" (tip 14), "at least 10 times faster" (tip 75), "more than 2 times faster" (tip 52), "almost 2 times faster" (tip 40), "10-50 times faster" (tip 86). The author declined to quantify these, citing variation between their own machines, and added a general caveat to the introduction naming the laptop and LAPACK version. That caveat helps, but the specific numbers still read as measured ratios rather than impressions. Replace each precise multiplier with an approximate form (for example "roughly twice as fast on our laptop") so the figure matches the evidence the authors say they have.
- [statistical-rigor] The two benchmark tables (tip 15 centering, tip 44 princomp vs prcomp) report six order statistics over 20 replications, with no paired difference between methods and no basis given for 20 runs. For princomp vs prcomp the medians differ by about 18 ms while the maxima nearly coincide (50.55 vs 51.37 ms), so a reader cannot judge how stable the ordering is. The author declined, preferring to present the output as the timing packages produce it. Report the paired difference and its spread for these two comparisons, and state that a single laptop with fixed JIT and BLAS settings limits how far the ratios carry.
- [prose-and-style] The introduction stacks three second-clause-after-comma sentences in the car/jeep paragraph ("One might observe …", "Differences from 40 seconds …", "Sometimes one tip saves …"); a reader skimming for the rule would be helped by tightening, though the passage is clear as written.
- [Reviewer 1] Phrase the bare ratio claims (tips 14, 40, 52, 75, 86) as observations on the authors' Windows laptop, consistent with the authors' own statement that the factors vary across machines.
- [Reviewer 3] Name the BLAS in the paper text, not only LAPACK 3.12.1, since the matrix timings depend on it.
- [Reviewer 3] The reproducible parallel RNG note (Section 9) is still absent. The decline rests on timing, but the point is reproducibility of simulation results.
- [methods-soundness] Several timing displays (Tips 16 and 46, the profiling output in Tip 2) report microbenchmark or Rprof results, but the data-generating lines and any seed are given only for some. Adding the generating code and a set.seed for each reported benchmark would let a reader reproduce the numbers. This was asked in the previous round and is still open.
- [Reviewer 2] Add collapse (fmean, fsum, fvar) to the grouped-operation tips (17, 38). This was asked in the previous round and is still open.
- [Reviewer 3] State the counter-case for the micro-optimizations that currently read as absolute (tip 42 `dim` vs `nrow`, tip 12 `mean` accuracy). This was asked in the previous round and is still open.

OPTIONAL:

- [statistical-rigor] A cross-machine, cross-BLAS comparison would separate ratios that are properties of R from ratios that are properties of this laptop. The author declined on scope grounds and notes that on an earlier machine prcomp was faster than princomp, which itself shows the ratios are not stable. A multi-platform table would settle which recommendations are backend-dependent, and can sit in a later paper.

We will correct these in production, and you confirm them in the proof:

- [writing-quality] Tip 17: "Note that thus command works also for matrices" should read "this command". The correct word is determined by the sentence, so this leaves the author's list and enters the proof.
- [writing-quality] Tip 19 (incremental calculations): "the first–rder partial correlations" is missing a letter and reads as "first-order". Confirm in the proof.

REVIEWS

Report Reviewer 1, recommends minor

Written by the journal's review engine in the role of the application reviewer, and read by the editor before the decision. How the engine works is on the peer review page.

Of the three required points and two lighter ones I raised, the tip count is reconciled (abstract, title and the list at 91 now agree), the R Inferno reference is supplied, tip 44 carries the princomp stability footnote, and tip 34 notes the NA difference between `x * (x > 0)` and `ifelse`. The content remains strong, and the numerical-overflow section (tips 77 to 85) is still the best part of the paper. The tip-17 rowsum alignment and the tip-82 underflow threshold, both raised by other reviewers, now read correctly.

Two things my prior report asked for are only partly settled. I asked for the BLAS to be stated and for the matrix comparisons to be qualified as backend-dependent. The authors added "The LAPACK version is 3.12.1" to the introduction and state that sessionInfo() is shipped with the code. LAPACK is not the BLAS, and the BLAS is what governs the relative cost of `crossprod`, `solve`, `eigen`, `princomp` and the `float` throughput tip. sessionInfo() does report the BLAS, so a reader who opens the script can find it, and the reproducibility minimum is therefore in hand. What is still missing is the one sentence that warns the reader of the paper, who never opens the script, that several definite orderings in Section 5 can shrink or invert on an optimised multi-threaded BLAS. The authors themselves note in their reply that on a different machine they once saw `prcomp` beat `princomp`, which is exactly the caveat that belongs in the text. I have marked this recommended because the shipped environment covers the verifiable minimum.

On the bare speed ratios (tips 14, 40, 52, 75, 86), the authors decline to quantify and explain that the factors differ between their own laptop and desktop. That is a reasonable decline for a tutorial of idioms. The cheap step that remains is to attach "on our machine" to these figures, which matches the authors' own account and costs nothing in content. Recommended.

The one point that still needs an action is the competing-interest statement. The paper recommends Rfast, Rfast2 and estats, which the authors maintain, and no declaration appears. The authors have asked the editor whether it is required and for help wording it. It is required, and placement is for the editor to confirm, so one workable form is: "M. Tsagris and M. Papadakis are the authors and maintainers of the R packages Rfast, Rfast2 and estats, which are recommended in this paper." This is new to my report, it was raised by two other reviewers, and it is the kind of point that holds a paper for one short exchange rather than another full round.

Numbered suggestions to the authors:

1. REQUIRED Add a competing-interest statement naming Rfast, Rfast2 and estats as packages the authors maintain. A workable sentence: "M. Tsagris and M. Papadakis are the authors and maintainers of the R packages Rfast, Rfast2 and estats, which are recommended in this paper." The editor will confirm placement.
2. RECOMMENDED Add one sentence to the introduction or Section 5 that the matrix comparisons were run with the BLAS on the described laptop, and that the margins for `crossprod` versus explicit trans-

pose-multiply, `princomp` versus `prcomp`, and the `float` tip can narrow or reverse under OpenBLAS or MKL.

3. RECOMMENDED Mark the bare ratio claims in tips 14, 40, 52, 75 and 86 as observations on the authors' Windows laptop, since the replies state the factors vary by machine.

Report Reviewer 2, recommends minor

Written by the journal's review engine in the role of the methodology reviewer, and read by the editor before the decision. How the engine works is on the peer review page.

Of the five required points and the lighter ones I raised, all but one are met, and the one that remains is the cheapest of the set. Tip 84 now describes underflow near 1e-308 with the subnormal range (tip 82 in the new numbering), the princomp stability cost is stated as a footnote to the matrix-PCA tip, the prcomp description now says singular value decomposition of the centered data, the n-1 rank statement for hd.eigen is corrected, and the tip count reads 91 consistently across the title, abstract and introduction. The accuracy cost of sum(x)/length(x) against mean is now in the text. The authored-package comparisons are framed in the introduction as comparisons against R's built-in functions, which is the softening route I offered, and I am satisfied with it. Section 10 adds matrixStats and collapse as complementary collections, which is a fair acknowledgement.

One required point is genuinely open. The competing-interest statement is still not in the manuscript. The author's reply asks the editor what to write and where, which is a reasonable process question, so I will answer it here rather than leave it hanging. The venue expects a short declaration wherever authors recommend packages they maintain, and the paper recommends Rfast, Rfast2, estats and hd.eigen in tips 62, 70, 72 and across Sections 9 and 10. A single sentence placed before the references settles it, for example: "The authors maintain the R packages Rfast, Rfast2 and estats, which are among the packages discussed in this paper." This is not a judgement on the recommendations, several of which are reasonable, only a disclosure of who wrote them.

The second required item is new in this version and I flag it as such. The author fixed the original tip 17 alignment problem for variant a2 by letting rowsum sort the groups, which is correct. Variant a3 then divides by as.vector(tabulate(ina)). tabulate returns one entry per integer position from 1 upward and inserts a zero for any absent code, so for a grouping like ina with values 1, 3, 4 it returns a length-4 vector while rowsum returns three rows. The division then recycles and misaligns silently. Tip 16 already warns that tabulate needs consecutive strictly positive values, and the same condition governs a3. A reader will copy this line, so either carry that condition into tip 17 or drop a3 and keep a2.

On reproducibility, the uploaded script now opens with sessionInfo and the environment is documented, and the declined seed is a reasonable call for a measurement of timing. The remaining gap is small. The introduction names the LAPACK version, which is a different library from the BLAS that drives crossprod, eigen and the float throughput tip. On default Windows R this is the reference BLAS, so one sentence saying the timings are reference-BLAS results on the described laptop and that the margins change under OpenBLAS or MKL would set reader expectations correctly. I have kept this recommended because the environment is now stated and a reader can reconstruct it.

Numbered suggestions to the authors:

1. REQUIRED Add a competing-interest statement before the references naming Rfast, Rfast2 and estats as packages the authors maintain, since the paper recommends them (tips 62, 70, 72, Sections 9 and 10). Suggested wording: "The authors maintain the R packages Rfast, Rfast2 and estats, which are among the packages discussed in this paper."
2. REQUIRED In tip 17, either add the condition that ina must be consecutive strictly positive integers for the a3 variant, or remove a3, because as.vector(tabulate(ina)) misaligns with rowsum(x, ina) when group codes are non-consecutive. This issue is new in this version.
3. RECOMMENDED Add one sentence to the introduction stating that the printed timings are reference-BLAS results on the described Windows laptop and that the margins vary with the linear algebra backend, since the stated LAPACK version does not fix the BLAS that governs the matrix tips.

Report Reviewer 3, recommends minor

Written by the journal's review engine in the role of the software reviewer, and read by the editor before the decision. How the engine works is on the peer review page.

Of the three required points and six lighter ones I raised, two of the three required are met and one is still open.

The tip 17 correctness bug is fixed. The a3 variant now reads `rowsum(x, ina) / as.vector(tabulate(ina))`, and the default `rowsum` sorts numeric group labels numerically, so for the consecutive positive-integer grouping that `tabulate` needs anyway the rows and counts line up. The tip count is now consistent across the title, abstract and introduction at 91, and the numbered list runs 1 to 91. The p-value underflow threshold in tip 82 is corrected to near 1e-308 with the subnormal range noted, the hd.eigen rank in tip 62 reads n-1, the princomp footnote in tip 44 states the SVD-versus-covariance accuracy cost, and tip 11 now states that `mean` makes two passes in extended precision. The data.table material in tip 17 adds grouped aggregation by reference, which covers the main gap I flagged. These all read correctly.

The competing-interest statement is the one required point that remains open. The paper recommends Rfast, Rfast2 and estats (tips 62, 70, 72 and Section 10), which the authors maintain, and there is no disclosure anywhere in the revised text. The authors deferred to the editor and asked for wording, which is a reasonable request, so a draft is in suggestion 1. This is a disclosure the venue expects when an authored package is recommended as faster, and it stays required until it appears in the manuscript.

On the reproducibility points I accept most of the reasoned declines. For timing comparisons a fixed seed genuinely does not matter, and the two printed tables (tip 15 centering, tip 44 princomp) can be reconstructed from the stated dimensions (10,000 by 20 and 10,000 by 50). Adding `sessionInfo()` at the top of the uploaded code is good provenance, since running it prints the linked BLAS and LAPACK. One gap stays worth a sentence. The intro names LAPACK 3.12.1, but the matrix timings (crossprod against explicit transpose-multiply, princomp against prcomp, the float tip) depend on the BLAS, which reference, OpenBLAS and MKL change by large factors. Naming the BLAS in the text, not only in the code, sets reader expectations. The parallel RNG note I asked for is still absent. The decline again rests on timing, but Section 9 motivates simulation studies, and a reader who parallelises one will want reproducible streams for the results, not the timings. It stays a recommendation and does not hold the paper.

Numbered suggestions to the authors:

1. REQUIRED Add a competing-interest statement. A suitable form: "M. Tsagris and M. Papadakis are the authors and maintainers of the R packages Rfast, Rfast2 and estats, which are recommended in this tutorial. These packages are free and open source and the authors derive no financial benefit from them." Place it as a short section before the references or in a footnote on first mention.
2. RECOMMENDED State the BLAS used for the benchmarks in the paper text, since the matrix timings in tips 44, 50 to 52 and 68 narrow or invert under OpenBLAS or MKL, and LAPACK 3.12.1 alone does not identify it.
3. RECOMMENDED In Section 9, add a note on reproducible parallel RNG using `clusterSetRNGStream` and `mc.set.seed`, aimed at reproducing simulation results rather than timings.

REVISION, RECEIVED 2026-11-09

Authors confirm each required point as done or give their reason for declining it. Further answers are voluntary, because the review compares every revision with the previous version and the next round's reports say what remains open.

We have addressed most of the reviewers' comments. Specifically, the set.seed argument is what we argue against. We have added the Section Conflicts of Interests, and uploaded the the updated script with the R code.

> REQUIRED The nearest prior work is the authors' own earlier preprint "Taking R to its limits: 70+ tips" (Tsagris and Papadakis, 2018, PeerJ Preprints, doi:10.7287/peerj.preprints.26605v1), which the search found and which does not appear in the reference list. The current manuscript states "to the best of our knowledge, this is the first time such a long list has been compiled in one place," and the response letter itself refers to "an earlier version of this paper, which can be found on the web." That earlier version already compiled a comparable list, so the first-compilation claim is not accurate as written without citing and positioning against it. Please cite the 2018

preprint and state what the present version adds relative to it (the count moving from 70+ to 91, the numerical-stability section, the data.table and matrix additions), or qualify the first-compilation sentence.

Done.

REQUIRED Tip 17 option a3 still misaligns in the same family of cases the prior round flagged. The author removed `reorder = FALSE`, so `rowsum(x, ina)` now returns groups in sorted order and divides by `as.vector(tabulate(ina))`. This agrees only when the group codes are the consecutive positive integers 1..k. When the codes have a gap, say `ina = c(1,3,3,1)`, `rowsum` returns two rows while `tabulate(ina)` returns `c(2,0,2)` of length three, and the division recycles, so each group mean is divided by the wrong count. Option a2 with `table(ina)` is safe because `table` returns only present levels in sorted order. Either drop a3, or add the "consecutive strictly positive integers" caveat that Tip 16 already states for `tabulate` and cross-reference it.

Done.

REQUIRED The manuscript still carries no competing-interest statement. The authors maintain Rfast, Rfast2 and estats, and the paper recommends these packages as faster in tips 62 (hd.eigen), 70 (Rfast::Dist), 72 (colMedians) and across Sections 9 and 10. Three reviewers asked for this in the prior round, and the response letter defers it ("We will ask the editors what to write and where") rather than adding it. The disclosure is required regardless of how the recommendations are judged. Add a short statement naming Rfast, Rfast2 and estats as packages the authors maintain, placed before the references. A usable form: "M. Tsagris and M. Papadakis are the maintainers of the R packages Rfast, Rfast2 and estats, which are recommended in this paper. The authors declare no other competing interests."

Done.

REQUIRED Add the competing-interest statement. It is still absent from the manuscript, and the paper recommends Rfast, Rfast2, estats and hd.eigen. A one-line declaration suffices and I give wording below.

Done.

REQUIRED State the BLAS in use and add a reproducibility script for the reported timings. Several matrix comparisons (crossprod vs t(X) %*% Y, princomp vs prcomp, float throughput) depend on which BLAS is linked, and reference BLAS results can invert under OpenBLAS or MKL. Provide the benchmark script plus sessionInfo and the BLAS identity, or qualify these comparisons as reference-BLAS results. This was asked in the previous round and is still open.

Done.

REQUIRED Where an authored package is recommended as faster, compare it against the strongest alternative, not base R alone. Either benchmark Rfast/estats functions against matrixStats, collapse and data.table, or soften the superiority claims to "an alternative to base R". This was asked in the previous round and is still open.

Done.

REQUIRED Ship a runnable benchmark script that regenerates the two printed tables (tips 16 and 46) with a fixed seed, the R version, and the BLAS named, or remove the printed numbers. The uploaded file reproduces code chunks but does not run end to end. This was asked in the previous round and is still open.

Done.

REQUIRED Add data.table coverage beyond `fread`: keyed joins, grouped aggregation with `[, by = ]`, and update by reference, cross-referenced from tips 12 and 17. This was asked in the previous round and is still open.

Done.

RECOMMENDED The reproducible parallel RNG note (Section 9) is still absent. The decline rests on timing, but the point is reproducibility of simulation results.

Declined. We argue against it because the goal is on the time comparisons, not the reproducibility of some random number that do not affect time.

RECOMMENDED Add collapse (fmean, fsum, fvar) to the grouped-operation tips (17, 38). This was asked in the previous round and is still open.

Declined. If we add them then we would have to add many more in other tips.

REQUIRED State the BLAS in use and add a reproducibility script for the reported timings. Several matrix comparisons (crossprod vs t(X) %*% Y, princomp vs prcomp, float throughput) depend on which BLAS is linked, and reference BLAS results can invert under OpenBLAS or MKL. Provide the benchmark script plus sessionInfo and the BLAS identity, or qualify these comparisons as reference-BLAS results.

Done.

> REQUIRED Add a competing-interest statement. The authors maintain Rfast, Rfast2 and estats, which the paper recommends as faster (tips 64, 72, Section 10).
>
> **Done.**

> REQUIRED Where an authored package is recommended as faster, compare it against the strongest alternative, not base R alone. Either benchmark Rfast/estats functions against matrixStats, collapse and data.table, or soften the superiority claims to "an alternative to base R".
>
> **Done.**

> REQUIRED Ship a runnable benchmark script that regenerates the two printed tables (tips 16 and 46) with a fixed seed, the R version, and the BLAS named, or remove the printed numbers. The uploaded file reproduces code chunks but does not run end to end.
>
> **Done.**

> REQUIRED Add data.table coverage beyond `fread`: keyed joins, grouped aggregation with `[, by = ]`, and update by reference, cross-referenced from tips 12 and 17.
>
> **Done.**

> RECOMMENDED Add collapse (fmean, fsum, fvar) to the grouped-operation tips (17, 38).
>
> **Declined.** If we add them then we would have to add many more in other tips.

Round 3: Accepted

SIGNED DECISION

Signed by Matthias Templ on 2026-11-14 · ORCID 0000-0002-8638-5276

Final acceptance of version 1, after the editor read the preview of the article. The journal ran the code end to end on Linux. Every printed comparison keeps its direction, and most speed-up factors are smaller than on the authors' laptop.

DECISION LETTER

Congratulations, your manuscript has been accepted for publication in JRepStat. Production begins now.

Where it is not on file yet, we still need the following for production.

- a valid ORCID iD for every author
- a captured computational environment (dependencies/lockfile/container)
- a signed determinism attestation for the entry-point command

REVIEWS

Report Reviewer 1, recommends accept

Written by the journal's review engine in the role of the application reviewer, and read by the editor before the decision. How the engine works is on the peer review page.

The one required point is settled. The new Conflicts of Interests section names Rfast, Rfast2 and estats as packages the authors maintain and declares no other competing interest, which is the statement I asked for.

My two recommended points are both addressed in substance. The reader-facing BLAS caveat is now in two places: the introduction states the BLAS version and that speed-up factors may vary across machines and operating systems, and tip 50 carries the specific warning that the crossprod margins can narrow or reverse under an optimized multithreaded BLAS. Tip 15 adds a matching note about fixed JIT and BLAS settings, and tip 44 now has the princomp precision footnote. On the bare ratios, tip 14 gained a footnote attributing the factor to the Windows laptop, and the authors declined to annotate every remaining ratio (tips 52, 75, 86) on the grounds that they would then have to add the same note across many tips. That decline convinces me: the global caveat in the introduction now covers the reader who meets any single ratio, so repeating it at each tip would add length without adding information.

Nothing remains open, and the paper reads cleanly as it now stands.

Numbered suggestions to the authors:

(none)

Report Reviewer 2, recommends accept

Written by the journal's review engine in the role of the methodology reviewer, and read by the editor before the decision. How the engine works is on the peer review page.

Of the three points I raised, all are met.

The competing-interest point is settled. A Conflicts of Interests section before the references states that both authors maintain Rfast, Rfast2 and estats and that these are recommended in the paper, which is the disclosure the venue expects.

The tip 17 alignment point is also settled. Variant a3 still divides rowsum(x, ina) by as.vector(tabulate(ina)), and the authors took the route of adding the condition rather than dropping a3. The new note tells the reader that tabulate returns frequencies only for consecutive strictly positive integers and should be treated with the caution of tip 16. That removes the silent-misalignment trap for a reader who copies the line.

The BLAS point, which I had kept recommended, is addressed more fully than I asked. The introduction now gives the BLAS version alongside LAPACK and states that speed-up factors may vary across machines and operating systems, and tips 15 and 50 add the backend caveat at the two places it matters most. A reader will now read the printed margins with the right expectations.

Nothing is open.

Numbered suggestions to the authors:

(none)

Report Reviewer 3, recommends accept

Written by the journal's review engine in the role of the software reviewer, and read by the editor before the decision. How the engine works is on the peer review page.

The single required point that remained open, the competing-interest disclosure, is now met. The new "Conflicts of Interests" section states that the authors maintain Rfast, Rfast2 and estats and that these are recommended in the paper, which is the disclosure the venue expects when an authored package is promoted as faster. That settles it, and nothing required remains.

On the parallel RNG note, the authors again declined on the ground that the goal is timing and not the reproducibility of draws that do not affect timing. The reason does not fully engage the point I raised, which was about reproducing the results of a simulation study a reader parallelises using the Section 9 recipe, not about the timings. It is a legitimate reasoned decline and it was a recommendation, so it does not hold the paper. I leave it there.

On the BLAS, the intro now reports "BLAS version is 4.61.24651.0" alongside LAPACK 3.12.1, and the uploaded script carries the session info. That is a partial answer. A version string does not identify whether the benchmarks ran against the single-threaded reference BLAS that ships with Windows R, or an optimized multithreaded backend, and the tips on crossprod, princomp and float narrow or invert across those. Naming the implementation would let a reader place the ratios. This is a one-word addition and does not hold the paper.

Numbered suggestions to the authors:

1. OPTIONAL In the intro, name the BLAS implementation used for the benchmarks (for example the reference BLAS shipped with Windows R), since the version string alone does not tell the reader whether the matrix timings came from a single-threaded or an optimized multithreaded backend.

APPENDIX A

The authors' full response letter to round 1

As uploaded with the revision received 2026-10-22. Its own pages follow.

Reviewer: - [consistency-check] The abstract and the introduction both state "more than 100 tips", but the body contains 94 numbered tips (1 through 94, the last being the `[[` versus `$` item in Section 11). The title says "90+ tips", which matches the count. Reconcile the abstract and the two introduction sentences ("a collection of more than 100 tips", "We have listed more than 100 tips") with the actual number so the claim agrees with the title and the content.

Reply: The typos is now fixed.

Reviewer: - [methods-soundness] Tip 64 states that prcomp() "performs the eigendecomposition of a matrix". prcomp() computes principal components by singular value decomposition of the (centered) data matrix, while princomp() uses the eigendecomposition of the covariance matrix. Please correct the description so the recommendation and its justification match what the functions do.

*Reply: We added this "*prcomp() "performs the eigendecomposition of a matrix". prcomp() computes principal components by singular value decomposition of the (centered) data matrix*".*

Reviewer: - [Reviewer 1] State the BLAS in use and add a reproducibility script for the reported timings. Several matrix comparisons (crossprod vs t(X) %*% Y, princomp vs prcomp, float throughput) depend on which BLAS is linked, and reference BLAS results can invert under OpenBLAS or MKL. Provide the benchmark script plus sessionInfo and the BLAS identity, or qualify these comparisons as reference-BLAS results.

Reply: We have uploaded the R code to reproduce the timing results. In the beginning of this code we have added the sessionInfo().

Reviewer: - [Reviewer 2] Add a competing-interest statement. The authors maintain Rfast, Rfast2 and estats, which the paper recommends as faster (tips 64, 72, Section 10).

Reply: We will ask the editors what action and what to write regarding this comment.

Reviewer: - [Reviewer 2] Where an authored package is recommended as faster, compare it against the strongest alternative, not base R alone. Either benchmark Rfast/estats functions

against matrixStats, collapse and data.table, or soften the superiority claims to "an alternative to base R".

Reply: We simply stated in the introduction that the comparisons are made with R's built-in functions.

Reviewer: - [Reviewer 2] Ship a runnable benchmark script that regenerates the two printed tables (tips 16 and 46) with a fixed seed, the R version, and the BLAS named, or remove the printed numbers. The uploaded file reproduces code chunks but does not run end to end.

Reply: We have updated the R code we upload now. We think we need these tables to show the time differences. We think the seed is not important, as we are measuring time. As mentioned earlier, the R code begins with the sessionInfo(), where information on the LAPACK (version 3.12.1) appears. In the tip #16 for example, we state that x contains 10,000 rows and 20 columns. So the user can generate a matrix of this size.

Reviewer: - [Reviewer 2] Correct tip 84: values below .Machine$double.eps (2.22e-16) are not rounded to zero. Underflow to zero happens near 1e-308 (and subnormals to 5e-324). The log.p advice stands, the stated threshold does not.

Reply: We changed this as the reviewer suggested.

Reviewer: - [Reviewer 3] Tip 17 option a3 divides `rowsum(x, ina, reorder = FALSE)` by `as.vector(table(ina))`, whose counts are in sorted group order, so the two are misaligned when groups do not first appear in sorted order. Fix the alignment or remove a3.

Reply: We removed the reorder=TRUE.

Reviewer: - [Reviewer 3] Add data.table coverage beyond `fread`: keyed joins, grouped aggregation with `[, by = ]`, and update by reference, cross-referenced from tips 12 and 17.

Reply: We added he command from data.table as well.

Reviewer: - [Reviewer 3] A short note on when to leave base R for Rcpp or RcppArmadillo yourself, not only via a wrapper package.

Reply: This depends upon whether the use knows how to write in C++ in an efficient manner. So, we cannot advise everyone to do that, even if they can program in C++.

Recommended:

Reviewer: - [methods-soundness] Most matrix timing claims (for example crossprod versus t(X) %*% Y in Tips 52 to 53, princomp versus prcomp in Tip 46, and the float and Matrix tips) depend on the BLAS/LAPACK backend and the platform. Under an optimized multithreaded BLAS several of these gaps shrink or reverse. One sentence stating that the benchmarks were run with the reference BLAS on the described laptop, and that margins vary with the linear algebra backend, would set the reader's expectations correctly.

Reply: We added this sentence in the introduction "In the uploaded \textit{R} code we have added the session info of the \textit{R} environment we used. The LAPACK version is 3.12.1."

Reviewer: - [methods-soundness] Tip 64 says at most n eigenvalues of the correlation matrix are non-zero when n < p. After centering and standardizing, the rank is at most n - 1, so at most n - 1 eigenvalues are non-zero. Adjust the statement, since the surrounding argument about hd.eigen() rests on this rank.

Reply: We fixed this typo.

Reviewer: - [methods-soundness] Several timing displays (Tips 16 and 46, the profiling output in Tip 2) report microbenchmark or Rprof results, but the data-generating lines and any seed are given only for some. Adding the generating code and a set.seed for each reported benchmark would let a reader reproduce the numbers.

Reply: We do not want to add a .seed because the generated data are of not importance, the timing is. Also, you know that the time results will vary among computers. So, we simply state our timings, and the R code is uploaded as an easy way for the reader to run the code, instead of copying from the pdf of the paper, they can copy the code from the script easier.

Reviewer: - [statistical-rigor] Several quantitative speed claims carry no measurement: "at least 10 times faster" for Newton-Raphson versus nlm (tip 77), "3-4 times faster" for the G2 tests (tip 15), "more than 2 times faster" for sum(x %*% m) (tip 54), "almost 2 times faster"

for dim over nrow (tip 42), and the many "much faster" statements in Section 10. Either give the timing that supports each figure or state the number as an approximate observation on the authors' machine.

Reply: These are personal experience, empirical evidence. Consider this as something we have seen in general.

Reviewer: - [statistical-rigor] The two benchmark tables (tips 16 and 46) report min, lower quartile, mean, median, upper quartile and max over 20 replications but no reasoning for 20 runs and no statement of the run-to-run spread of the difference. For prcomp versus princomp the maxima (50.55 and 51.37 ms) nearly coincide while the medians differ by about 18 ms, so a reader cannot tell how stable the ordering is. Report the paired difference and its spread, and note that a single laptop with JIT and BLAS settings fixed limits how far the ratios generalise.

Reply: We report these because the R packages that measure time they give this type of output. We did not want to go deep in the comparison, memory for example, or what the reviewer suggests.

Reviewer: - [writing-quality] In the introduction, "Gillespie and Lovelace (2016) gave a bookâ€“length treatment in his book" refers to two authors, so "his" should read "their". The sentence also repeats "book" three times (Wickham "in his book", Burns "in his book The R Inferno", "a bookâ€“length treatment in his book"); one pass to vary the wording would read more cleanly.

Reply: We corrected this typo. We also proof read the whole manuscript again, after we completed the changes.

Reviewer: - [writing-quality] Tip 80 contains a spelling error, "afraind", that reads as an author typo rather than an extraction artefact. Please correct to "afraid".

Reply: We modified this sentence, and removed the verb "afraid" completely.

Reviewer: - [Reviewer 1] Quantify a representative set of the bare "faster" claims (for example tips 42, 54, 89) with the same benchmark script, so the factors ("almost 2 times", "more than 2 times faster", "10-50 times") are supported.

Reply: We cannot quantify because our observations differ between a laptop and a desktop. We cannot even quantify between our laptops, as we see different speed-ups.

Reviewer: - [Reviewer 2] Add collapse (fmean, fsum, fvar) to the grouped-operation tips (17, 38).

Reply: As we said in an earlier comment, and added in the Introduction, our comparisons take place against R's built-in functions. But, we do mention alternative packages as well.

Reviewer: - [Reviewer 3] Note the numerical-stability cost of `princomp` over `prcomp` in tip 46, consistent with the paper's own stability section.

Reply: We did not understand what the reviewer is trying to say with this comment.

Reviewer: - [Reviewer 3] Add a note on reproducible parallel RNG (`clusterSetRNGStream`, `mc.set.seed`) in Section 9, given the simulation use case the paper motivates.

Reply: We insist on not giving a .seed number, because the goal is to see the timings, not to reproduce exactly the same computations and mathematical results. The user may run these commands and see for themselves.

Reviewer: - [Reviewer 3] State the counter-case for the micro-optimizations that currently read as absolute (tip 42 `dim` vs `nrow`, tip 12 `mean` accuracy).

Reply: We did not understand what the reviewer is trying to say with their comment.

Optional:

Reviewer: - [methods-soundness] Tip 46 recommends princomp over prcomp on speed grounds, while the numerical-overflow section stresses stability. princomp via the covariance eigendecomposition is less accurate for ill-conditioned data than the SVD used by prcomp. A short note on when the speed gain is safe would align the two parts of the paper.

Reply: We were not aware of the stability issues to be honest and thank you for this. The goal of this paper is about mainly about speed. To us, what your are saying sounds like R's issue, something that R should fix.

Reviewer: - [statistical-rigor] A comparison across more than one machine, operating system and BLAS backend would show which of the reported ratios are properties of R and which are properties of this particular laptop.

Reply: When writing this paper we opted against this strategy, because perhaps more issues would arise and the paper would be lengthier. We list our tips, which we must highlight may not work across all laptops and as R evolves some may not work in the future. In the past for example and using another laptop we had seen ***that prcomp was faster than princomp****. In an earlier version of this paper, which can be found on the web, you will notice that we mention this.*

We will correct these in production, and you confirm them in the proof:

Reviewer: - [consistency-check] The profiling output in Tip 2 lists a function `"h"` (self.time 0.01) that does not appear in the function `f` shown just above it. The self-times still sum correctly (8.56 + 0.84 + 0.01 = 9.41), so the total is internally consistent, but the `"h"` row has no source in the displayed code. Confirm the intended output.

Reply: We ran it again and now present the new output.

Reviewer: - [writing-quality] Section 10 has a dropped package name: "The same is true for ordinal logistic regression in and in ordinal (Christensen, 2015)". A package name is missing before the first "in" (the comparison names only one of the two packages being compared). Please supply the intended package.

Reply: We corrected this typo.

Reviewer: - [Reviewer 1] The R Inferno (Patrick Burns) is cited in the introduction but has no entry in the references. Supply the reference.

Reply: We added the reference of the book.

Reviewer 1

****Recommendation:** minor**

This is a useful tutorial. The problem is real: R users who run simulations or matrix-heavy analyses routinely leave large speed and memory gains on the table, and a single well-organised list of concrete idioms has clear value for that audience. The coverage is broad and mostly accurate. The numerical-overflow section (tips 79 to 88) is the strongest part. The inverse-logit reformulation, the softplus formula max(x,0) + log1p(exp(-abs(x))), the log-sum-exp trick, working in log-densities, and using log.p for sortable tail probabilities are all correct and are exactly the things practitioners get wrong. The matrix-algebra section is also solid: crossprod/tcrossprod, solve(X, Y) over solve(X) %*% Y, backsolve/forwardsolve for triangular factors, and the associativity point in tip 61 are all sound and well chosen.

Reviewer: My main concern as an application reviewer is that many speed claims are asserted without evidence, and the few that carry benchmarks were run on a single laptop with an unstated BLAS. This matters because a good share of the matrix tips are BLAS-bound. Whether crossprod(X, Y) beats t(X) %*% Y, and by how much, and whether princomp beats prcomp, depends on whether R is linked against reference BLAS, OpenBLAS, or MKL. On an optimised multi-threaded BLAS some of these gaps shrink or reverse. Readers will apply these tips on machines unlike yours, so the paper should say which BLAS produced the numbers and should ship the script that produced tables 16 and 46. That also lets the quantified factors scattered through the text ("almost 2 times faster", "more than 2 times faster", "10-50 times faster") be checked. A short reproducibility script with sessionInfo would settle all of this at once and is the natural minimum for a tutorial whose value is its timings.

Reply: We have uploaded the R code and the sessionInfo() in the beginning of the code. The statements 10-50 times faster were observed in our laptops. Perhaps with a very strong desktop, and Linux instead of Windows, the speed-ups are smaller.

Reviewer: One consistency point deserves attention because the paper itself raises the standard. Section 7 states that one should not sacrifice numerical stability for speed. Tip 46

then recommends princomp over prcomp on speed grounds alone. princomp works from the covariance matrix and its eigendecomposition, which loses precision relative to the SVD of the centered data that prcomp uses, and the difference shows up on ill-conditioned or nearly collinear data that applied users meet often. A single sentence flagging the tradeoff keeps the tip while honouring the paper's own principle. A smaller correctness note: tip 36 gives x * (x > 0) for ifelse(x > 0, x, 0), which differs in the presence of NA, and tip 87's integer point is right and could add that length/dim/nrow return integers so n * n overflows silently, which the example already shows.

Reply: We were not aware of this limitation of princomp and so we added this sentence as a footnote in tip 46: "Note however, that \texttt{princomp} works from the covariance matrix and its eigendecomposition, which loses precision relative to the SVD of the centered data that \texttt{prcomp} uses, and the difference shows up on ill--conditioned or nearly collinear data that applied users might meet."

Reviewer: The headline count needs fixing. The abstract and introduction say "more than 100" tips, the title says "90+", and the list runs to 94. The claim that this is the first compilation of such length depends on getting that number straight, so please set it to what the paper actually contains. The R Inferno is discussed in the introduction but is absent from the references and should be added. On presentation, the tips read as a flat sequence within each section, and a few (38 on apply for medians, 48 on by) qualify their own advice in ways that could confuse a reader skimming for a rule. A one-line summary table grouping the tips by theme would help the tutorial's usability, though the paper stands without it.

Reply: We corrected the typo regarding the 100+ tips. An earlier version of the paper consisted of more than 70 tips. We expanded them to 100+ and then removed some of them, this is why this disagreement occurred, which we should have corrected. Bear in mind that these tips are based on our personal experience with our laptops using Windows.

Numbered suggestions to the authors:

Reviewer: 1. [required] Set the tip count consistently across the title, abstract, and introduction to the number actually present (94 by my count), since the "first such list" claim rests on it.

Reply: We now state in the Introduction that we have listed 94 tips.

Reviewer: 2. [required] State the BLAS used for the benchmarks and provide the script plus sessionInfo that produced the timing tables (tips 16 and 46). Note that crossprod vs explicit transpose-multiply, princomp vs prcomp, and the float throughput claim are BLAS-dependent and can narrow or invert under OpenBLAS or MKL, so present them as reference-BLAS results or qualify them.

Reply: We state in the Introduction the Lapack version used in our session.

Reviewer: 3. [required] Add one sentence to tip 46 noting that princomp is less numerically stable than prcomp because it eigendecomposes the covariance matrix, so the speed gain trades against precision on ill-conditioned data.

Reply: We added it.

Reviewer: 4. [editor_fix] Add a references entry for The R Inferno (P. Burns), which is cited in the introduction but missing from the reference list.

Reply: We added the reference.

Reviewer: 5. [recommended] Support the unquantified speed factors for a representative subset (for example tips 42, 54, and 89) using the same benchmark script, so the stated multipliers are backed by numbers.

Reply: These numbers are based on our experience and may not be observed with different machines or operating systems. What we want to stress is that these recommendations are faster, even to the slightest.

Reviewer: 6. [recommended] Note in tip 36 that x * (x > 0) and ifelse(x > 0, x, 0) differ when x contains NA, so readers do not swap them blindly.

Reply: We added this information under the chunk of the code.

Reviewer: 7. [optional] A short table grouping the tips by theme (memory, vectorization, matrix algebra, numerical stability) would make the tutorial easier to navigate and reference.

Reply: We like the grouping we have done, and that is why you see them organized by those sections.

Reviewer 2

****Recommendation:** major**

This is a tutorial, and I review it as one. A curated list of idioms does not owe a benchmark for every line the way a methods paper owes one for its headline result. The numerical-overflow section (tips 79 to 88) is the strongest part: the inverse-logit reformulation, the softplus identity max(x,0)+log1p(exp(-|x|)), the log-sum-exp shift, log1p and expm1, and working with log-densities are all correct and well chosen. The vectorization and preallocation advice is sound and the worked profiling example makes its point cleanly.

Reviewer: My concerns are methodological and they cluster. First, the paper is written by the maintainers of Rfast, Rfast2 and estats, and it recommends those packages as faster (tips 64 and 72, and Section 10). There is no competing-interest statement, and that needs to be added. Tied to it is a fairness question that runs through the paper. Many tips present a fast option against the slowest baseline only. Where the recommended option is an authored package, the relevant comparison is against the strongest alternative in the ecosystem, which for column medians, variances and distances means matrixStats, collapse and data.table, and for a truncated eigendecomposition (tip 64) means something like RSpectra or irlba. Please either benchmark against those or phrase the recommendation as an alternative to base R. I have marked the comparison required with the softening as the low-cost route.

Reply: Since the paper was based initially on comparing the three R packages you mentioned to the built-in R functions we decided to leave it as is, but mention alternative packages that offer fast executions as well. Regarding the statement, we will ask the editors to tell us what to add and where.

Reviewer: Second, reproducibility of the numbers you print. The paper shows two benchmark tables (centering in tip 16, princomp against prcomp in tip 46) and asserts ratios elsewhere ("more than 2 times faster", "almost 2 times faster"). None of the matrix timings can be read without the BLAS, since reference BLAS, OpenBLAS and MKL change the relative cost of crossprod, solve, eigen and the float tip (70) by large factors. The environment gives the R version but not the BLAS, and the code file cannot regenerate the two tables because

several chunks depend on objects defined only in the prose. A single script that builds the inputs, sets a seed, runs microbenchmark or bench, and prints the two tables would close this, or the printed numbers could be removed and the claims kept qualitative.

Reply: In the uploaded R code script you will see our sessionInfo(). There you will see the LAPACK version installed in our R environment.

Reviewer: Third, two technical corrections. Tip 84 states that p-values below .Machine$double.eps are rounded to zero. That is off by roughly 290 orders of magnitude. A p-value of 1e-100 is representable and sortable. Underflow to zero happens near the smallest normal double, about 1e-308, with subnormals below that. The advice to carry log p-values is good, so only the justification needs fixing. Tip 46 recommends princomp over prcomp for speed. princomp forms and diagonalises the covariance matrix, which is the less stable route, while prcomp uses the SVD of the centered data. In a paper whose own motto is not to trade stability for speed, that recommendation needs one sentence of caveat. In the same spirit, tip 12 recommends sum(x)/length(x) over mean for speed, and mean's extra cost buys the two-pass accuracy you are giving up, which is worth a half-sentence.

Reply: We have corrected the tip 84 now, and added a few worda about tip 46. Regarding tip 12, the accuracy is mentioned.

Reviewer: A lighter point on the many micro-claims. "nrow is almost 2 times faster than dim(X)[1]" (tip 42), the internal .colSums (tip 39), iterating over elements (tip 93): these are true as ratios and negligible as absolute times. A reader can spend real effort chasing nanoseconds. Where you give a ratio without a measurement, either measure it or mark it as a constant-factor nicety.

Reply: It is true, the savings is measured in nanoseconds. But, for someone who performs millions of simulation studies every week, the nanoseconds become seconds or minutes with time. The message we want to pass is that one should make light-speed fast functions whenever possible. If a model requires 0.01 seconds to fit once, when it comes to millions of repeats, the time cost is high.

Numbered suggestions to the authors:

Reviewer: 1. [required] Add a competing-interest statement naming Rfast, Rfast2 and estats as packages the authors maintain, since the paper recommends them (tips 64, 72, Section 10).

Reply: We will ask the editors what to do with this, how to write the statement.

Reviewer: 2. [required] For each authored-package recommendation, compare against the strongest alternative (matrixStats, collapse, data.table, and RSpectra or irlba for tip 64), or reword the claim to "an alternative to base R" where you do not benchmark.

Reply: Our initial decision was to compare against the built-in functions for illustration purposes. Overall, the goal is to show the users how to write efficient code. To avoid the comparison we mention a few packages and encourage authors to search for fast alternatives in packages.

Reviewer: 3. [required] Supply a runnable script that regenerates the tables in tips 16 and 46 with a fixed seed, the R version, and the BLAS stated, or remove the printed timings and keep the claims qualitative.

Reply: We have uploaded the updated R code with the session info included where we show the LAPACK version in our R environment.

Reviewer: 4. [required] In tip 84, replace the .Machine$double.eps threshold with the correct underflow behaviour near 1e-308 and the subnormal range, keeping the log.p recommendation.

Reply: We fixed this.

Reviewer: 5. [required] Add one sentence to tip 46 that princomp diagonalises the covariance matrix and is less stable than the SVD-based prcomp, so the speed gain carries an accuracy cost.

Reply: We added a sentence regarding the computational instability the princomp may face in contrast to the more robust prcomp.

Reviewer: 6. [editor_fix] Make the tip count consistent: the abstract says "more than 100", the title says "90+", and the list ends at 94.

Reply: We fixed this typo.

Reviewer: 7. [recommended] Add collapse (fmean, fsum, fvar) to the grouped-operation tips 17 and 38, where it is typically faster than rowsum and aggregate.

Reply: We preferred to compare our functions against the R's built-in functions. We do mention this and other packages in our manuscript.

Reviewer: 8. [recommended] Temper the unquantified ratio claims (tips 42, 54, 70, 93) to qualitative statements, or back each with a measurement, and add a half-sentence to tip 12 on the accuracy that sum/length gives up relative to mean.

Reply: The speed-ups are based on our personal experience, and vary between laptops and desktops, this is why we do not report the exact numbers.

Reviewer: 9. [optional] A table of the matrix-tip speedups under reference BLAS, OpenBLAS and MKL would settle how much each recommendation depends on the linear algebra backend.

Reply: We have LAPACK in our R environment.

Reviewer 3

I read this as a software and reproducibility reviewer, with the uploaded R file treated as part of the submission. The tutorial is useful and most of the advice is correct and well chosen. The matrix-algebra tips (crossprod, solve(X, Y), backsolve, the Mahalanobis route to the multivariate normal exponent) and the numerical-overflow section (log-sum-exp, log1p, log-densities, log p-values) are the strongest parts and are the kind of thing many R users genuinely do not know. My concerns are about whether the shipped artifact does what the submission record says it does, one correctness bug, and a disclosure that a computing venue expects.

Reviewer: The declared entry point is `Rscript R code.R`, which signals a runnable artifact. It is not one. The file's own header states that several fragments refer to objects defined only in surrounding prose, and a top-to-bottom run fails early: chunk 2 calls `microbenchmark(mean(x), ...)` with no `x` defined, chunk 5 uses `n`, chunk 6 uses `fast` and `slow`, and so on. No `set.seed` appears anywhere, so even the self-contained chunks that build data with `rnorm` do not reproduce fixed numbers. The two timing tables in the paper (centering in tip 16, princomp against prcomp in tip 46) are pasted into the code as comments and are not regenerated by any runnable, seeded code. The machine is named, which is good provenance, but timings also depend on the BLAS, the OS and the R build, none of which is pinned beyond the prose. For a paper whose entire value rests on speed claims, a reader should be able to run something and see the numbers appear. The cheapest honest fix is to split the file: keep the illustrative snippets as clearly non-executable, and add two or three small seeded scripts that build the inputs and print the benchmark tables. If the file is meant to be snippets only, the declared entry point should say so.

Reply: We prefer to give the script with all snippets of code that appear in the manuscript. The reviewer has a point, x does not appear, but we mention the length or dimensions of x (if matrix) in the manuscript. The updated uploaded R code.R script includes now the sessionInfo() where the user can see that we use LAPACK in our R environment.

Reviewer: Tip 17 has a correctness bug in its "a bit faster" variant. `rowsum(x, ina, reorder = FALSE)` returns group rows in order of first appearance, while `as.vector(table(ina))` returns counts in sorted factor-level order. When the two orders differ, each group mean is divided by another group's count. The default `rowsum(x, ina)` in a2 is fine because it sorts. Please

either align the counts to the `reorder = FALSE` order or drop a3. This belongs in a tutorial precisely because a reader will copy it.

Reply: We fixed this typo.

Reviewer: The authors maintain Rfast, Rfast2 and estats, and the paper recommends these packages in tips 64, 72 and 74 and across Sections 9 and 10. A reader is entitled to know that. Please add a short interest declaration. This is not a judgement on the recommendations, several of which are reasonable, only on disclosing who wrote the packages being recommended.

Reply: We will ask the editors what to write and where to place it.

Reviewer: Two smaller content points are worth lining up with the paper's own stated priorities. Section 7 argues that numerical stability should not be sacrificed for speed, yet tip 46 recommends `princomp` over `prcomp` purely on timing. `princomp` forms the covariance and eigendecomposes it, which is less accurate for near-collinear data than the SVD route in `prcomp`, and one sentence should say so. On data.table, the paper uses it only through `fread`. For the aggregation tips (12 and 17) and for large grouped work generally, `[, by = ]`, keyed joins and update by reference are often the fastest route available, and a tutorial on efficient R aimed at large data reads as incomplete without them. The tip count in the abstract and introduction ("more than 100") does not match the 94 numbered tips or the "90+" in the title.

Reply: We have resolved these issues.

Numbered suggestions to the authors:

Reviewer: 1. [required] Make `R code.R` run end to end with inputs defined and `set.seed` set, or re-declare it as non-executable snippets and add separate seeded scripts that regenerate the centering (tip 16) and princomp/prcomp (tip 46) tables.

Reply: We prefer to avoid the .seed number, as we are not interested in the mathematical results, but on the timing results.

Reviewer: 2. [required] Fix tip 17: align `rowsum(x, ina, reorder = FALSE)` with counts in the same order, or remove the a3 variant, because dividing by `as.vector(table(ina))` uses sorted-order counts.

Reply: We fixed this inconsistency.

Reviewer: 3. [required] Add a declaration of the authors' interest in Rfast, Rfast2 and estats where those packages are recommended.

Reply: We will ask the editor's opinion on how to address this.

Reviewer: 4. [editor_fix] Reconcile the tip count: abstract and introduction say "more than 100", the list ends at 94, and the title says "90+".

Reply: We fixed it.

Reviewer: 5. [recommended] Add data.table beyond `fread`: keyed joins, grouped aggregation with `[, by = ]`, and update by reference, linked from tips 12 and 17.

Reply: We did not understand the reviewer's comment here.

Reviewer: 6. [recommended] In tip 46, state that `princomp` is less accurate than `prcomp` for near-collinear data.

Reply: We added this information.

Reviewer: 7. [recommended] In Section 9, add a note on reproducible parallel RNG using `clusterSetRNGStream` and `mc.set.seed`.

Reply: We prefer not to have a .seed number, as our prime interest lies on time, not on reproducing the same mathematical computations.

Reviewer: 8. [recommended] Give the counter-case for the micro-optimizations that read as absolute, including tip 42 (`dim` vs `nrow`) and the accuracy cost in tip 12 (`mean` vs `sum(x)/length(x)`).

Reply: These are based on personal experience, so the seed-ups may differ, and we prefer not to state them.

Reviewer: 9. [optional] Add a short note on when to move to Rcpp or RcppArmadillo yourself, as a complement to the wrapper-package route.

Reply: This depends upon whether the use knows how to write in C++ in an efficient manner. So, we cannot advise everyone to do that, even if they can program in C++.