

Commentary



Comments on “Implementation of Miettinen-Nurminen score method with or without stratification in R” by Lee MH and Bae KS (2022)

Peter J. Laud

Statistical Services Unit, University of Sheffield, Sheffield S3 7RH, UK

OPEN ACCESS

► See the article “Implementation of Miettinen-Nurminen score method with or without stratification in R” in volume 30 on page 155.

Received: Jun 27, 2025
Revised: Aug 7, 2025
Accepted: Aug 19, 2025
Published online: Sep 16, 2025

*Correspondence to

Peter J. Laud

Statistical Services Unit, University of Sheffield,
Hicks Building, Hounsfield Road, Sheffield S3
7RH, UK.

Email: p.j.laud@sheffield.ac.uk

Copyright © 2025 Translational and Clinical
Pharmacology

It is identical to the Creative Commons
Attribution Non-Commercial License (<https://creativecommons.org/licenses/by-nc/4.0/>).

ORCID iDs

Peter J. Laud
<https://orcid.org/0000-0002-3766-7090>

Conflict of Interest

- Authors: Nothing to declare
- Reviewers: Nothing to declare
- Editors: Nothing to declare

Lee and Bae’s recent article [1] presented an implementation of the Miettinen-Nurminen (MN) asymptotic score method [2], which is provided within the extensive sasLM package (reference version 0.10.5 at the time of writing) for R [3]. The authors incorrectly claimed that “The MN score method has not been previously implemented in R software for data with stratification.” In fact, the stratified MN method for risk difference (RD), relative risk (RR) and odds ratio (OR) has been available in the ratesci package [4] (from the CRAN repository <https://cran.r-project.org/package=ratesci>) since 2016, via the `scoreci()` function with the arguments `stratified = TRUE`, `weighting = "MN"`, `bctf = TRUE`, `skew = FALSE`. The package has always contained clear documentation of the stratified MN score method within the published repository, so it is not clear how the authors were unable to discover it—I can only speculate that the contents of the documentation may have lacked specific terms for a limited internet search. It is helpful that another implementation of the method is now available for cross-checking, but unfortunately the outputs of the two packages do not match, due to errors in the sasLM source code.

Since publication of the article, the authors have updated the unstratified `RRmn1()` function, removing MN’s variance bias correction [2] (which I denote as an ‘N-1’ correction) (compare the code on page 159 of the article with the updated version at <https://github.com/cran/sasLM/releases/tag/0.10.5>). Presumably this was to resolve the discrepancy against the PropCIs package [5] highlighted in the article. However, achieving consistency with PropCIs comes at the cost of inconsistency within the sasLM package: the stratified `RRmn()` function retains the ‘N-1’ correction, as do the RD and OR functions. Note that MN included the correction for all three contrasts, so any method labelled as “MN” (or “mn”) should include it. The same inconsistency exists in the PropCIs package, and it is not clear why the ‘N-1’ correction is omitted only for RR. It seems that it is absent in Nam’s non-iterative formula [6] cited in the PropCIs documentation. Note that the uncorrected score method for RR was first published by Koopman [7], which Gart and Nam [8] demonstrated was identical to MN apart from the ‘N-1’ correction, and further work by Gart and Nam [9] (adding stratification) omitted the variance correction, focussing instead on correcting for skewness. Without the ‘N-1’ correction, `RRmn1()` gives a Koopman interval. I urge the sasLM authors (who are not reliant on the Nam formula) to reinstate it in the `RRmn1()` code to reflect the MN methodology referenced in the package documentation.

Table 1. Estimated treatment effects for 6/10 vs. 6/20 with unstratified MN intervals for RD, RR and OR, from sasLM functions, compared with `ratesci::scoreci()`, and SAS PROC FREQ (discrepancies highlighted)

Software	Contrast (sasLM function)		
	RD (RDmn1)	RR (RRmn1)	OR (ORmn1)
sasLM	-0.07397, 0.60673	0.84354, 4.59408 [†]	0.73546, 16.68497
<code>ratesci::scoreci()</code> *	-0.0739619777, 0.6067195463	0.8309741988, 4.6579915649	0.7354663814, 16.6849625037
SAS PROC FREQ	-0.0739619777, 0.6067195463 [‡]	0.8309741988, 4.6579915647	0.7354663814, 16.6849625036

MN, Miettinen-Nurminen; RD, risk difference; RR, relative risk; OR, odds ratio.

*With arguments: stratified = FALSE, bcf = TRUE, skew = FALSE, or_bias = FALSE, contrast = "RD", "RR" or "OR".

[†]RRmn1() produces the Koopman method, not MN.

[‡]With increased precision using the option CL=SCORE(CONVERGE=1E-12).

Aside from the 'N-1' correction for RR, I have confirmed that the unstratified sasLM functions produce reasonably precise results (within the limited default tolerance set in the `uniroot()` function), by cross-checking various examples against output from `ratesci::scoreci()`, SAS[®] PROC FREQ and other packages (one such example is shown in Table 1, using an example dataset from Gart & Nam [8]). However, for stratified calculations, there are some errors in the sasLM implementation that can produce very unexpected results for RR, and smaller discrepancies for RD. Stratified intervals for OR are accurate when they are produced, but the point estimate is not, and there are conditions that cause `ORmn()` to fail altogether. I have reviewed the sasLM source code and identified the cause of these discrepancies, which I have submitted to the package maintainers directly by email. Details of the coding errors are included in Appendix 1.

To illustrate the problem for RR, note that the reported stratified estimates of p_1 and p_2 (labelled as R_1 and R_2 in the article) and the selected contrast (in the `$Common` element of the output list) should be consistent with each other, i.e. the point estimate for RR should equal p_1/p_2 . For the authors' example dataset (p. 160-1) from a trial of Molnupiravir for the treatment of COVID-19, the point estimate for the common RR is given as 0.7172, but $p_1/p_2 = 0.0677/0.0973 = 0.6959$. In datasets with more diverse RR estimates between strata, the effect of this discrepancy can be severe: it is possible for the common RR to be estimated by `sasLM::RRmn()` with a confidence interval entirely above $RR = 1$ when $p_1/p_2 < 1$. A similar issue affects the point estimate for OR, causing the `ORmn()` function to fail, and it also fails if any stratum has $y_2 = 0$ or $y_1 = n_1$.

Table 2 summarises the estimates and stratified MN intervals for all 3 contrasts using the Molnupiravir trial data, showing the differences between sasLM and `ratesci` output (and also SAS Viya[®] PROC FREQ [10] for RD). Note that the 95% interval for RR from `sasLM::RRmn()` contains unity, but the `ratesci::scoreci()` version does not. The CMH test for this dataset (from `mantelhaen.test()` without continuity correction, or from the /CMH option of SAS PROC FREQ) gives $\chi^2 = 4.0712$, $p = 0.0436$, and the RD confidence

Table 2. Estimated treatment effects for Molnupiravir vs. Placebo (data from Lee and Bae's Table 3 [1]) with stratified Miettinen-Nurminen intervals for RD, RR and OR, from sasLM functions, compared with `ratesci::scoreci()`, and SAS Viya PROC FREQ for RD (discrepancies highlighted)

Software	Contrast (sasLM function)		
	RD (RDmn)	RR (RRmn)	OR (ORmn)
sasLM	-0.0296 (-0.0590, -0.0012)	0.7172 (0.5016, 1.0248)	0.6910 (0.4591, 0.9889)
<code>ratesci::scoreci()</code> *	-0.0296 (-0.0589, -0.0009)	0.6959 (0.4887, 0.9898)	0.6737 (0.4590, 0.9889)
SAS Viya PROC FREQ	-0.0296 (-0.0589, -0.0009)	-	-

RD, risk difference; RR, relative risk; OR, odds ratio.

*With arguments: stratified = TRUE, weighting = "MN", bcf = TRUE, skew = FALSE, or_bias = FALSE, contrast = "RD", "RR" or "OR".

interval excludes $RD = 0$, so the sasLM version of the RR confidence interval would make a substantial difference to the conclusion of the study and contradict the test for association.

I have also validated `ratesci::scoreci()` against various alternative sources using other weighting schemes, including: SAS macro %SCORECI [11] (Mantel-Haenszel (MH) and Inverse Variance (INV) weighting for all contrasts); `metalite.ae::rate_compare()` (MH weighting for RD); an archived copy of the PF package (RR with Gart-Nam efficient score formulation, matching INV weights from ratesci); and the `strat.MHRD.MN()` published by Klingenberg [12] (RD with MH weighting).

Since the authors raised the limited precision of `PropCIs::orscoreci()`, it is also worth mentioning that sasLM does not offer a huge improvement in this regard. The `uniroot()` function has a default convergence tolerance which (on my system at least) is approximately 0.0001, so the results only match to around 4 decimal places (Table 1). This might be sufficient for most reporting purposes, but it would be helpful to some users (and for code validation) to allow an increased level of precision to be specified (e.g., by adding a function argument that modifies the 'tol' argument of `uniroot()`). The authors also claimed that SAS uses the same limited procedures as PropCIs for root-finding, but I have not found the results from SAS to lack precision. The output window only displays 4 decimal places, but an ODS OUTPUT statement can be used to save the results in a dataset, which can be reported with greater precision by applying an appropriate numeric format. For unstratified RR and OR, `ratesci::scoreci()` matches output of SAS PROC FREQ to around 9 decimal places, which appears to be the maximum precision available in SAS. For unstratified RD, precision can be increased in SAS by specifying a smaller convergence criterion, for example with `CL=SCORE (CONVERGE=1E-12)`, but the same option does not appear to affect RR and OR output. The `precis` argument for `ratesci::scoreci()` controls the precision of output for all contrasts.

In addition to addressing or avoiding the above issues, the ratesci package offers further improvements to the score-based methods, with options to include Gart and Nam's skewness correction (which improves one-sided coverage to give centrally located intervals [13]), and an additional bias correction for the OR score [14,15]. It also provides *p*-values for the 2-sided test for association and for one-sided non-inferiority tests against a specified null parameter value, as well as a heterogeneity test. Furthermore, analysis of Poisson rates is also catered for using the same score methodology, which can be useful for the analysis of exposure-adjusted adverse event incidence rates.

Finally, noting that the purpose of the sasLM package is to provide SAS-like functionality in R, it is also worth pointing out that the stratified score methods are not currently available from SAS PROC FREQ in SAS v9.4 (SAS/STAT 15.3). They have recently been implemented in the SAS Viya platform (with the `COMMONRISKDIFF (CL=MN)` option for the TABLES statement), but only for the RD contrast. A SAS macro implementation of the score confidence intervals for all contrasts, including optional skewness correction, non-inferiority and heterogeneity tests, and Poisson contrasts, is available at <https://github.com/petelaud/ratesci-sas>. Currently the %SCORECI macro (version 0.2.0) offers MH and INV weighting, but implementation of MN weighting is planned for a future update.

REFERENCES

1. Lee MH, Bae KS. Implementation of Miettinen-Nurminen score method with or without stratification in R. *Transl Clin Pharmacol* 2022;30:155-162. [PUBMED](#) | [CROSSREF](#)
2. Miettinen O, Nurminen M. Comparative analysis of two rates. *Stat Med* 1985;4:213-226. [PUBMED](#) | [CROSSREF](#)
3. Bae KS. sasLM: 'SAS' linear model. R package version 0.10.5 [Internet]. <https://CRAN.R-project.org/package=sasLM>. Accessed June 23, 2025.
4. Laud P. ratesci: confidence intervals and tests for comparisons of binomial proportions or Poisson rates. R package version 1.0.0 (2025) [Internet]. <https://cran.r-project.org/package=ratesci>. Accessed June 23, 2025.
5. Scherer R. PropCIs: various confidence interval methods for proportions. R package version 0.3-0 [Internet]. <https://CRAN.R-project.org/package=PropCIs>. Accessed June 23, 2025.
6. Nam J. Confidence limits for the ratio of two binomial proportions based on likelihood scores: non-iterative method. *Biom J* 1995;37:375-379. [CROSSREF](#)
7. Koopman PAR. Confidence limits for the ratio of two binomial proportions. *Biometrics* 1984;40:513-517. [CROSSREF](#)
8. Gart JJ, Nam J. Approximate interval estimation of the ratio of binomial parameters: a review and corrections for skewness. *Biometrics* 1988;44:323-338. [PUBMED](#) | [CROSSREF](#)
9. Gart JJ, Nam JM. Approximate interval estimation of the difference in binomial parameters: correction for skewness and extension to multiple tables. *Biometrics* 1990;46:637-643. [PUBMED](#) | [CROSSREF](#)
10. SAS Institute Inc. SAS® Viya® platform programming documentation [Internet]. https://documentation.sas.com/doc/en/pgmsascdc/v_044/statug/statug_freq_syntax08.htm#statug.freq.freqcommonriskdiff. Accessed June 23, 2025.
11. Laud P. ratesci-sas: confidence intervals and tests for comparison of rates in SAS [Internet]. <https://github.com/petelaud/ratesci-sas>. Accessed June 23, 2025.
12. Klingenberg B. A new and improved confidence interval for the Mantel-Haenszel risk difference. *Stat Med* 2014;33:2968-2983. [PUBMED](#) | [CROSSREF](#)
13. Laud PJ. Equal-tailed confidence intervals for comparison of rates. *Pharm Stat* 2017;16:334-348. [PUBMED](#) | [CROSSREF](#)
14. Gart JJ. Analysis of the common odds ratio: corrections for bias and skewness. *Bull Int Stat Inst* 1985;45:175-176.
15. Laud PJ. Equal-tailed confidence intervals for comparison of rates. *Pharm Stat* 2018;17:290-293. [PUBMED](#) | [CROSSREF](#)

Appendix 1

Coding errors in sasLM

This appendix includes further details of the errors in the sasLM (version 0.10.5) source code leading to discrepancies when comparing against results produced by the ratesci package and SAS PROC FREQ.

Note that the greatest magnitude of discrepancy in **Table 2** (resulting in a material change in the conclusion of the study) is for the relative risk (RR) contrast, so I will focus on that first.

With reference to equation (17) and associated definition of r_i^* in Miettinen-Nurminen (MN) 1985, and noting that the `Obj()` function within `sasLM: :RRmn()` is defined as a function of a variable named `rr` (which is called RR in the MN paper), a corrected version of the score definition in the `Obj()` subroutine would use:

```
r1s = sum(w * p1)
r2s = sum(w * p2)
v = (r1 * (1 - r1)/n1 + rr * rr * r2 * (1 - r2)/n2) *
    (n1 + n2)/(n1 + n2 - 1)
return((r1s - r2s * rr)^2/sum(w * w * v) - v0)
```

Similarly, from MN's equation (18), the weight defined in `wrr()` should be:

$$w = 1/((1 - r1s)/((1 - r2s))/n1 + rr/n2)$$

Furthermore, the `sasLM: :RRmn()` code derives a stratified point estimate, 'RR' (not to be confused with the 'RR' in the MN paper), by applying the weights to the per-stratum estimates for RR:

$$RR = \text{sum}(w2/\text{sum}(w2) * p1 / p2)$$

With this formulation, it is possible for `sasLM: :RRmn()` to obtain a very contradictory set of results, where the common RR is estimated with a confidence interval entirely above RR=1 while the point estimate is < 1 (**Table A1**).

A somewhat better point estimate could be calculated as the ratio of weighted estimates of p_1 and p_2 as follows:

$$RR = \text{sum}(w2/\text{sum}(w2) * p1) / \text{sum}(w2/\text{sum}(w2) * p2)$$

This appears to match the maximum likelihood estimate (MLE) for RR (the point at which the stratified score is zero [8]) when MN weights are used, but that is not the case if an inverse variance weighting scheme is used.

For the odds ratio (OR) contrast, the point estimate is more problematic. `sasLM: :ORmn()` uses a weighted combination of the OR in each stratum, which fails if any stratum has $y_2 = 0$ or $y_1 = n_1$, because the stratum estimate of OR is infinite. The function also fails in situations with diverse stratum OR estimates (such as the example in Table A1), where

the crudely pooled OR estimate can fall outside the confidence interval, resulting in an error “f() values at end points not of opposite sign” from the uniroot function.

This weighted OR estimate is not the MLE with any weighting scheme. The modified OR estimate from weighted estimates of p1 and p2 might be ‘close enough’ in most circumstances, but would not be immune to failure. A more reliable general approach, for any contrast with any weighting, is to obtain the MLE by identifying the contrast parameter value that produces a stratified score equal to zero. The setup of the sasLM functions presents difficulties doing this with the uniroot() function, because the chi-squared score function is not monotonic. The ratesci package avoids this difficulty by using the corresponding z-statistic instead.

The above errors do not affect the RDmn() function, but small discrepancies arise because of the following lines in the p1p2rd() subroutine, which should use pmax() and pmin() instead of max() and min():

```
p = sign(q)*sqrt(pmax(0, L2^2/(3*L3)^2 - L1/(3*L3)))
a = (pi + ifelse(p == 0, acos(0), acos(pmin(1, pmax(-1, q/p^3)))/3
```

Table A1. Example dataset with diverse stratum estimates for RR and OR, with stratified results from sasLM and ratesci functions

Stratum	Active treatment	Placebo	Stratum RR	Stratum OR
1	4/20	8/20	0.5	0.384
2	2/20	11/20	0.182	0.097
3	10/20	2/20	5	8.494
sasLM function			RRmn()	ORMn()
sasLM result: crude weighted point estimate			1.894	2.992
sasLM result: estimate implied by estimates of common p1 = 0.2666, p2 = 0.35			0.762	0.675
sasLM result: CI			(1.053, 3.597)	“f() values at end points not of opposite sign”
ratesci::scoreci(): MLE & CI			0.762 (0.438, 1.309)	0.682 (0.317, 1.464)

RR, relative risk; OR, odds ratio; MLE, maximum likelihood estimate; CI, confidence interval.