National Institute for Health and Care Excellence

Blood transfusion (update)

NICE guideline NG24

Methods

February 2026

FINAL

Disclaimer

The recommendations in this guideline represent the view of NICE, arrived at after careful consideration of the evidence available. When exercising their judgement, professionals are expected to take this guideline fully into account, alongside the individual needs, preferences and values of their patients or service users. The recommendations in this guideline are not mandatory and the guideline does not override the responsibility of healthcare professionals to make decisions appropriate to the circumstances of the individual patient, in consultation with the patient and/or their carer or guardian.

Local commissioners and/or providers have a responsibility to enable the guideline to be applied when individual health professionals and their patients or service users wish to use it. They should do so in the context of local and national priorities for funding and developing services, and in light of their duties to have due regard to the need to eliminate unlawful discrimination, to advance equality of opportunity and to reduce health inequalities. Nothing in this guideline should be interpreted in a way that would be inconsistent with compliance with those duties.

NICE guidelines cover health and care in England. Decisions on how they apply in other UK countries are made by ministers in the Welsh Government, Scottish Government, and Northern Ireland Executive. All NICE guidance is subject to regular review and may be updated or withdrawn.

Copyright

ISBN: 978-1-4731-9319-2

Contents

Development of the guideline

What this guideline covers

This guideline update discusses:

- Tranexamic acid for surgery to reduce the need for blood transfusion for adults, children and young people

What this guideline does not cover

This guideline did not review evidence on the following:

- alternatives to blood transfusion (other than tranexamic acid)
- red blood cells
- platelets
- fresh frozen plasma
- cryoprecipitate
- prothrombin complex concentrate
- patient safety
- patient information
- blood transfusions for patients with acute upper gastrointestinal bleeding.

These sections were reviewed in the original guideline by methods from NICE, the manual (2014). Their recommendations will be retained in this update, although they may have been revised to update language, reflect current policy or practice or to ensure consistency.

Methods

This evidence review was developed using the methods and process described in Developing NICE guidelines: the manual.

Declarations of interest were recorded according to NICE's conflicts of interest policy.

Developing the review questions and outcomes

The two review questions developed for this guideline were based on the key areas identified in the guideline scope. They were drafted by the NICE guideline development team and refined and validated by the guideline committee.

The review questions were based on the following frameworks:

- Population, Intervention, Comparator, Outcome and Study type (PICOS) for reviews of interventions

Full literature searches, critical appraisals and evidence reviews were completed for all review questions.

Reviewing research evidence

Review protocols

Review protocols were developed with the guideline committee to outline the inclusion and exclusion criteria used to select studies for each evidence review. Where possible, review protocols were prospectively registered in the PROSPERO register of systematic reviews.

The protocol for review A was taken from Jaiswal et al, 2025. This can be found on PROSPERO. The protocol for review B was not registered on PROSPERO or the NICE website due to the review being started before the committee had finalised the protocol in order to ensure the timely completion of the work.

Searching for evidence

Evidence was searched for each review question using the methods specified in Developing NICE guidelines: the manual.

Selecting studies for inclusion

All references identified by the literature searches and from other sources (for example, previous versions of the guideline or studies identified by committee members) were uploaded into EPPI reviewer software (version 5) and de-duplicated. Titles and abstracts were assessed for possible inclusion using the criteria specified in the review protocol. At least 10% of the abstracts were reviewed by two reviewers, with any disagreements resolved by discussion or, if necessary, a third independent reviewer.

The full text of potentially eligible studies was retrieved and assessed according to the criteria specified in the review protocol. A standardised form was used to extract data from included studies. Study investigators were contacted for missing data when time and resources allowed (when this occurred, this was noted in the evidence review and relevant data was included).

Incorporating published evidence syntheses

If published evidence syntheses were identified sufficiently early in the review process (for example, from the surveillance review or early in the database search), they were considered for use as the primary source of data, rather than extracting information from primary studies. Syntheses considered for inclusion in this way were quality assessed to assess their suitability using the appropriate checklist, as outlined in Table 1. Note that this quality assessment was solely used to assess the quality of the synthesis in order to decide whether it could be used as a source of data, as outlined in Table 2, not the quality of evidence contained within it, which was assessed in the usual way as outlined in the section on 'Appraising the quality of evidence'.

Table 1: Checklists for published evidence syntheses

Type of synthesis	Checklist for quality appraisal
Systematic review of quantitative evidence	ROBIS
Network meta-analysis	Modified version of the PRISMA NMA tool (see appendix K of 'Developing NICE guidelines, the manual')
Qualitative evidence synthesis	ENTREQ reporting standard for published evidence synthesis is the generic reporting standard for QES, however specific reporting standards exist for meta-ethnography (eMERGe) and for realist synthesis (RAMESES II). If these reporting standards are not appropriate to the QES then an adapted PRISMA framework is used (see Flemming K, Booth A, Hannes K, Cargo M, Noyes J. Cochrane Qualitative and Implementation Methods Group guidance series-paper 6: reporting guidelines for qualitative, implementation, and process evaluation evidence syntheses. Journal of Clinical Epidemiology 2018; 97: 79-85).
Individual patient data meta-analysis	Checklist based on Tierney, Jayne F., et al. "Individual participant data (IPD) meta-analyses of randomised controlled trials: guidance on their use." PLoS Med 12.7 (2015): e1001855.

Each published evidence synthesis was classified into one of the following three groups:

- High quality – It is unlikely that additional relevant and important data would be identified from primary studies compared to that reported in the review, and unlikely that any relevant and important studies have been missed by the review.

- Moderate quality – It is possible that additional relevant and important data would be identified from primary studies compared to that reported in the review, but unlikely that any relevant and important studies have been missed by the review.
- Low quality – It is possible that relevant and important studies have been missed by the review.

Each published evidence synthesis was also classified into one of three groups for its applicability as a source of data, based on how closely the review matches the specified review protocol in the guideline. Studies were rated as follows:

- Fully applicable – The identified review fully covers the review protocol in the guideline.
- Partially applicable – The identified review fully covers a discrete subsection of the review protocol in the guideline (for example, some of the factors in the protocol only).
- Not applicable – The identified review, despite including studies relevant to the review question, does not fully cover any discrete subsection of the review protocol in the guideline.

The way that a published evidence synthesis was used in the evidence review depended on its quality and applicability, as defined in Table 2. When published evidence syntheses were used as a source of primary data, data from these evidence syntheses were quality assessed and presented in GRADE tables in the same way as if data had been extracted from primary studies. In questions where data was extracted from both systematic reviews and primary studies, these were checked to ensure none of the data had been double counted through this process.

Table 2: Criteria for using published evidence syntheses as a source of data

Quality	Applicability	Use of published evidence synthesis
High	Fully applicable	Data from the published evidence synthesis were used instead of undertaking a new literature search or data analysis. Searches were only done to cover the period of time since the search date of the review. If the review was considered up to date (following discussion with the guideline committee and NICE lead for quality assurance), no additional search was conducted.

High	Partially applicable	Data from the published evidence synthesis were used instead of undertaking a new literature search and data analysis for the relevant subsection of the protocol. For this section, searches were only done to cover the period of time since the search date of the review. If the review was considered up to date (following discussion with the guideline committee and NICE lead for quality assurance), no additional search was conducted. For other sections not covered by the evidence synthesis, searches were undertaken as normal.
Moderate	Fully applicable	Details of included studies were used instead of undertaking a new literature search. Full-text papers of included studies were still retrieved for the purposes of data analysis. Searches were only done to cover the period of time since the search date of the review.
Moderate	Partially applicable	Details of included studies were used instead of undertaking a new literature search for the relevant subsection of the protocol. For this section, searches were only done to cover the period of time since the search date of the review. For other sections not covered by the evidence synthesis, searches were undertaken as normal.

Methods of combining evidence

Data synthesis for intervention studies

Where possible, meta-analyses were conducted to combine the results of quantitative studies for each outcome. Network meta-analyses were considered in situations where there were at least 3 treatment alternatives and sufficient studies to make this possible. No network meta-analyses were conducted for this guideline update. When there were 2 treatment alternatives, pairwise meta-analysis was used to compare interventions.

Pairwise meta-analysis

Pairwise meta-analyses were performed in Cochrane RevMan (Review Manager) version 9.12.0, with the exception of ratio of means analyses which were carried out in R version 4.5.1 using the package ‘metafor’. A pooled relative risk was calculated for dichotomous outcomes (using the Inverse Variance or Mantel–Haenszel method) reporting numbers of people having an event, and a pooled incidence rate ratio was calculated for dichotomous outcomes reporting total numbers of events. Both relative and absolute risks were presented, with absolute risks calculated by applying the relative risk to the risk in the comparator arm of the meta-analysis (calculated as the total number events in the comparator arms of studies in the meta-analysis divided by the total number of participants in the comparator arms of studies in the meta-analysis).

A pooled mean difference was calculated for continuous outcomes (using the inverse variance method) when the same scale was used to measure an outcome across different studies. Where different studies presented continuous data measuring the same outcome but using different numerical scales (e.g. a 0-10 and a 0-100 visual analogue scale), these outcomes were all converted to the same scale before meta-analysis was conducted on the mean differences.

Where outcomes measured the same underlying construct but used different instruments/metrics, data were analysed using standardised mean differences (SMDs, Hedges’ g), as implemented in Cochrane RevMan. Alternative approaches to standardisation described in the NICE technical support unit guideline methodology

document on meta-analysis of continuous outcomes were used when this was needed for consistency with a network meta-analysis.

For continuous outcomes analysed as mean differences, change from baseline values were used in the meta-analysis if they were accompanied by a measure of spread (for example standard deviation). Where change from baseline (accompanied by a measure of spread) were not reported, the corresponding values at the timepoint of interest were used. If only a subset of trials reported change from baseline data, final timepoint values were combined with change from baseline values to produce summary estimates of effect. For continuous outcomes analysed as standardised mean differences this was not possible. In this case, if all studies reported final timepoint data, this was used in the analysis. If some studies only reported data as a change from baseline, analysis was done on these data, and for studies where only baseline and final time point values were available, change from baseline standard deviations were estimated, assuming a correlation coefficient derived from studies reporting both baseline and endpoint data, or if no such studies were available, assuming a correlation of 0.5 as a conservative estimate (Follman et al. 1992; Fu et al. 2013). In cases where SMDs were used they were usually converted back to a single scale to aid interpretation by the committee where possible, and when it was considered useful for decision making.

Random effects models were fitted when significant between-study heterogeneity in methodology, population, intervention or comparator was identified by the reviewer in advance of data analysis. This decision was made and recorded before any data analysis was undertaken. For all other syntheses, fixed- and random-effects models were fitted, with the presented analysis dependent on the degree of heterogeneity in the assembled evidence. Fixed-effects models were the preferred choice to report, but in situations where the assumption of a shared mean for fixed-effects model were clearly not met, even after appropriate pre-specified subgroup analyses were conducted, random-effects results are presented. Fixed-effects models were deemed to be inappropriate if there was significant statistical heterogeneity in the meta-analysis, defined as I^2 more than or equal to 50%.

However, in cases where the results from individual pre-specified subgroup analyses were less heterogeneous (with I^2 less than 50%) the results from these subgroups

were reported using fixed effects models. This may have led to situations where pooled results were reported from random-effects models and subgroup results were reported from fixed-effects models.

Where sufficient studies were available, meta-regression was considered to explore the effect of study level covariates.

Appraising the quality of evidence

Intervention studies (relative effect estimates)

RCTs and quasi-randomised controlled trials were quality assessed using the Cochrane Risk of Bias Tool. Non-randomised controlled trials and cohort studies were quality assessed using the ROBINS-I tool. Other study types (for example controlled before and after studies) were assessed using the preferred option specified in the NICE guidelines manual (appendix H). Evidence on each outcome for each individual study was classified as proposed by tool developers should be used where available, without change of wording.

Each individual study was also classified into one of three groups for directness, based on if there were concerns about the population, intervention, comparator and/or outcomes in the study and how directly these variables could address the specified review question. Studies were rated as follows:

- Direct – No important deviations from the protocol in population, intervention, comparator and/or outcomes.
- Partially indirect – Important deviations from the protocol in one of the following areas: population, intervention, comparator and/or outcomes.
- Indirect – Important deviations from the protocol in at least two of the following areas: population, intervention, comparator and/or outcomes.

Minimally important differences (MIDs) and clinical decision thresholds

The Core Outcome Measures in Effectiveness Trials (COMET) database was searched to identify published minimal clinically important difference thresholds relevant to this guideline that might aid the committee in identifying clinical decision thresholds for the purpose of GRADE. Identified MIDs were assessed to ensure they had been developed and validated in a methodologically rigorous way, and were applicable to the populations, interventions and outcomes specified in this guideline. In addition, the Guideline Committee were asked to prospectively specify any outcomes where they felt a consensus clinical decision threshold could be defined from their experience. In particular, any questions looking to evaluate non-inferiority

(that one treatment is not meaningfully worse than another) required a clinical decision threshold to be defined to act as a non-inferiority margin.

The committee agreed that for any questions where dichotomous outcomes were present, a 25% margin of difference around the line of null effect would be allowed for the measurement of precision. This would be indicated by a risk ratio confidence interval range of 0.8-1.25.

Clinical decision thresholds were used to assess imprecision using GRADE and aid interpretation of the size of effects for different outcomes. Clinical decision threshold that were used in the guideline are reported in the relevant evidence reviews.

GRADE for intervention studies analysed using pairwise analysis

GRADE was used to assess the quality of evidence for the outcomes specified in the review protocol. Data from randomised controlled trials, non-randomised controlled trials and cohort studies (which were quality assessed using the Cochrane risk of bias tool or ROBINS-I) were initially rated as high quality. The quality of the evidence for each outcome was downgraded or not from this initial point, based on the criteria given in Table 3. These criteria were used to apply preliminary ratings, but were overridden in cases where, in the view of the analyst or committee the uncertainty identified was unlikely to have a meaningful impact on decision making.

Table 3: Rationale for downgrading quality of evidence for intervention studies

GRADE criteria	Reasons for downgrading quality
Risk of bias	• Not serious (don't downgrade): less than 50% overall weighting some concerns/high risk of bias • Serious (downgrade 1 level): more than 50% some concerns/high risk of bias. • Very serious (downgrade 2 levels): more than 50% high risk of bias.
Indirectness	• Not serious (don't downgrade): less than 50% of overall weighting partially direct or indirect. • Serious (downgrade 1 level): more than 50% of overall weighting partially direct or indirect. • Very serious (downgrade 2 levels): more than 50% of overall weighting indirect

GRADE criteria	Reasons for downgrading quality
Inconsistency	Concerns about inconsistency of effects across studies, occurring when there is unexplained variability in the treatment effect demonstrated across studies (heterogeneity), after appropriate pre-specified subgroup analyses have been conducted. This was assessed using the I^2 statistic. • Not serious (don't downgrade) I^2 = less than 40%; • Serious (downgrade 1 level) I^2 = 40-60%; • Very serious (downgrade 2 levels) I^2 = more than 60%.
Imprecision	Where established MIDs are available these were used. If there are no established MIDs imprecision was assessed using Optimal Information Size and event rates. Interpretation of imprecision and consideration of baseline event rates was discussed with the committee whenever feasible and a summary included in the CDE section of each evidence review. If an MID other than the line of no effect was defined for the outcome, the outcome was downgraded once if the 95% confidence interval for the effect size crossed one line of the MID, and twice if it crosses both lines of the MID. If the line of no effect was defined as an MID for the outcome, it was downgraded once if the 95% confidence interval for the effect size crossed the line of no effect (i.e. the outcome was not statistically significant), and twice if the sample size of the study was sufficiently small that it is not plausible any realistic effect size could have been detected. Outcomes meeting the criteria for downgrading above were not downgraded if the confidence interval was sufficiently narrow that the upper and lower bounds would correspond to clinically equivalent scenarios. If Optimal Information Size was used: • outcomes were downgraded once if the sample size was lower than 100% of expected but greater than 30% and twice if the sample size was smaller than 30% of the Optimal Information Size • where the proportion of the confidence intervals for dichotomous outcomes exceeded 3 for risk ratios or 2.5 for odds ratios, the outcome was downgraded twice • where the total sample size for continuous outcomes exceeded 800, the outcome was not downgraded for imprecision.
Publication bias	Where 10 or more studies were included as part of a single meta-analysis, a funnel plot was produced to graphically assess the potential for publication bias. When a funnel plot showed convincing evidence of publication bias, or the review team became aware of other evidence of publication bias (for example, evidence of unpublished trials where there was evidence that the effect estimate differed in published and unpublished data), the outcome was downgraded once. If no evidence of publication bias was found for any outcomes in a review (as was often the case), this domain was excluded from GRADE profiles to improve readability.

GRADE automation in Power BI

GRADE analysis was performed in Microsoft Power BI using the outputs from Review Manager. Power M query details are included in appendices A and B for each review.

Reviewing economic evidence

Inclusion and exclusion of economic studies

No original health economic literature review seeking to identify published cost–utility analyses was conducted for this guideline update. Instead, at scoping it was decided that the primary health economic work for the Blood Transfusion guideline update would involve assessing the health economic analysis produced as part of new identified evidence that promoted this guideline update – the National Institute for Health and Care Research (NIHR) funded work produced by Jaiswal et al. 2025. An economic evidence profile, including critical appraisal according to the Guidelines manual, was completed for this included study.

Appraising the quality of economic evidence

The cost-utility analysis conducted as part of the work produced by Jaiswal et al. 2025 was appraised using a methodology checklist designed for economic evaluations (NICE guidelines manual; 2014). This checklist is not intended to judge the quality of a study per se, but to determine whether an existing economic evaluation is useful to inform the decision-making of the committee for a specific topic within the guideline.

There are 2 parts of the appraisal process. The first step is to assess applicability (that is, the relevance of the study to the specific guideline topic and the NICE reference case); evaluations are categorised according to the criteria in Table 4.

Table 4 Applicability criteria

Level	Explanation
Directly applicable	The study meets all applicability criteria, or fails to meet one or more applicability criteria but this is unlikely to change the conclusions about cost effectiveness
Partially applicable	The study fails to meet one or more applicability criteria, and this could change the conclusions about cost effectiveness
Not applicable	The study fails to meet one or more applicability criteria, and this is likely to change the conclusions about cost effectiveness. These studies are excluded from further consideration

In the second step, only those studies deemed directly or partially applicable are further assessed for limitations (that is, methodological quality); see categorisation criteria in Table 5.

Table 5 Methodological criteria

Level	Explanation
Minor limitations	Meets all quality criteria, or fails to meet one or more quality criteria but this is unlikely to change the conclusions about cost effectiveness
Potentially serious limitations	Fails to meet one or more quality criteria and this could change the conclusions about cost effectiveness
Very serious limitations	Fails to meet one or more quality criteria and this is highly likely to change the conclusions about cost effectiveness. Such studies should usually be excluded from further consideration

Where relevant, a summary of the main findings from the systematic search, review and appraisal of economic evidence is presented in an economic evidence profile alongside the clinical evidence.

Health economic modelling

No original health economic modelling was undertaken for the Blood Transfusion guideline update.

References

Follmann D, Elliott P, Suh I, Cutler J (1992) Variance imputation for overviews of clinical trials with continuous response. Journal of Clinical Epidemiology 45:769–73

Fu R, Vandermeer BW, Shamliyan TA, et al. (2013) Handling Continuous Outcomes in Quantitative Synthesis In: Methods Guide for Effectiveness and Comparative Effectiveness Reviews [Internet]. Rockville (MD): Agency for Healthcare Research and Quality (US); 2008-. Available from: http://www.ncbi.nlm.nih.gov/books/NBK154408/

Jaiswal N, Robinson W, Ciminata G et al. (2025) At what levels of expected blood loss from surgery is tranexamic acid (TXA) effective at reducing the need for blood transfusion?

Norman G., Sloan JA., Wyrwich KW. (2003) Interpretation of changes in health-related quality of life: the remarkable universality of half a standard deviation. Med Care 41(5):582-92.

Appendix A Review [A] Power M query for GRADE automation

Data rows

```
let
    Source = Excel.Workbook(Web.Contents("data source"), null, true),
    #"Data rows_Sheet" = Source{[Item="Data rows",Kind="Sheet"]}[Data],
    #"Promoted Headers" = Table.PromoteHeaders(#"Data rows_Sheet",
[PromoteAllScalars=true]),
    #"Changed Type" = Table.TransformColumnTypes(#"Promoted
Headers",{{"Analysis group", Int64.Type}, {"Analysis number", Int64.Type}, {"Analysis
name", type text}, {"Subgroup", type any}, {"Applicability", type text}, {"Study", type
text}, {"Study year", Int64.Type}, {"GIV Mean", type number}, {"GIV SE", type
number}, {"Experimental mean", type number}, {"Experimental SD", type number},
{"Experimental cases", Int64.Type}, {"Experimental N", Int64.Type}, {"Control mean",
type number}, {"Control SD", type number}, {"Control cases", Int64.Type}, {"Control
N", Int64.Type}, {"O-E", type any}, {"Variance", type any}, {"Weight", type number},
{"Mean", type number}, {"CI start", type number}, {"CI end", type number},
{"Footnotes", type any}}),
    #"Splitting by "" at "" to separate follow up time" = Table.SplitColumn(#"Changed
Type", "Analysis name", Splitter.SplitTextByDelimiter(" at ", QuoteStyle.Csv),
{"Analysis name.1", "Analysis name.2"}),
    #"Changed Type1" = Table.TransformColumnTypes(#"Splitting by "" at "" to
separate follow up time",{{"Analysis name.1", type text}, {"Analysis name.2", type
text}}),
    #"Renamed Columns" = Table.RenameColumns(#"Changed Type1",{{"Analysis
name.2", "Follow up time"}}),
    #"Splitting by "" ("" to separate scale information" = Table.SplitColumn(#"Renamed
Columns", "Analysis name.1", Splitter.SplitTextByDelimiter(" (", QuoteStyle.Csv),
{"Analysis name.1.1", "Analysis name.1.2"}),
    #"Changed Type2" = Table.TransformColumnTypes(#"Splitting by "" ("" to
separate scale information",{{"Analysis name.1.1", type text}, {"Analysis name.1.2",
type text}}),
```

```
    #"Renamed Columns1" = Table.RenameColumns(#"Changed Type2",{{"Analysis name.1.2", "Scale"}, {"Analysis name.1.1", "Protocol Outcome"}}),
    #"Removing closing bracket" = Table.ReplaceValue(#"Renamed Columns1",")","",Replacer.ReplaceText,{"Scale"}),
    #"Making short outcome" = Table.AddColumn(#"Removing closing bracket", "Outcome short", each if [Protocol Outcome] = "Proportion of patients requiring transfusion" then "Number Transfusions" else if [Protocol Outcome] = "All-cause mortality" then "Mortality" else if [Protocol Outcome] = "Length of stay" then "Length of stay" else if [Protocol Outcome] = "Number of units of allogenic blood transfused" then "Units Transfused" else if [Protocol Outcome] = "Volume of allogenic blood transfused" then "Volume Transfused" else if [Protocol Outcome] = "Serious adverse events" then "Serious AE" else if [Protocol Outcome] = "Thrombotic complications" then "Thrombosis" else if [Protocol Outcome] = "Acute myocardial infarction" then "MI" else if [Protocol Outcome] = "Infection" then "Infection" else "?"),
    #"Merged Queries" = Table.NestedJoin(#"Making short outcome", {"Study"}, BiasInd, {"Study"}, "BiasInd", JoinKind.LeftOuter),
    #"Expanded Critical appraisal no edits" = Table.ExpandTableColumn(#"Merged Queries", "BiasInd", {"Short population", "Short comparison"}, {"Short population", "Short comparison"}),
    #"Removed Duplicates" = Table.Distinct(#"Expanded Critical appraisal no edits"),
    #"Removed Columns" = Table.RemoveColumns(#"Removed Duplicates",{"Footnotes"}),
    #"Inserted Merged Column" = Table.AddColumn(#"Removed Columns", "GRADE number", each Text.Combine({Text.From([Analysis group], "en-GB"), Text.From([Analysis number], "en-GB")}, "-"), type text),
    #"Risk intervention" = Table.AddColumn(#"Inserted Merged Column", "Risk Intervention", each [Experimental cases] / [Experimental N], type number),
    #"Risk control" = Table.AddColumn(#"Risk intervention", "Risk control", each [Control cases] / [Control N], type number),
    #"Risk ratio" = Table.AddColumn(#"Risk control", "Risk ratio", each [Risk Intervention] / [Risk control], type number),
    #"1/a" = Table.AddColumn(#"Risk ratio", "1/a", each Number.Power([Experimental cases], -1), Int64.Type),
```

```
    #"1/c" = Table.AddColumn(#"1/a", "1/c", each Number.Power([Control cases], -1),
Int64.Type),
    #"a+b" = Table.AddColumn(#"1/c", "a+b", each [Experimental cases] +
[Experimental N], Int64.Type),
    #"c+d" = Table.AddColumn(#"a+b", "c+d", each [Control cases] + [Control N],
Int64.Type),
    #"1/a+b" = Table.AddColumn(#"c+d", "1/a+b", each Number.Power([#"a+b"], -1),
type number),
    #"1/c+d" = Table.AddColumn(#"1/a+b", "1/c+d", each Number.Power([#"c+d"], -1),
type number),
    #"Inserted Natural Logarithm" = Table.AddColumn(#"1/c+d", "ln(RR)", each
Number.Ln([Risk ratio]), type number),
    #"Inserted Addition" = Table.AddColumn(#"Inserted Natural Logarithm", "1/a +
1/c", each [#"1/a"] + [#"1/c"], Int64.Type),
    #"Inserted Subtraction" = Table.AddColumn(#"Inserted Addition", "Subtraction",
each [#"1/a + 1/c"] - [#"1/a+b"], type number),
    #"Inserted Subtraction1" = Table.AddColumn(#"Inserted Subtraction",
"Subtraction.1", each [Subtraction] - [#"1/c+d"], type number),
    #"Inserted Square Root" = Table.AddColumn(#"Inserted Subtraction1", "Square
Root", each Number.Sqrt([Subtraction.1]), type number),
    #"1.96*SQRT" = Table.AddColumn(#"Inserted Square Root", "1.96*SQRT", each
[Square Root] * 1.96, type number),
    #"ln(RR)+1.96*SQRT" = Table.AddColumn(#"1.96*SQRT", "Addition", each
[#"ln(RR)"] + [#"1.96*SQRT"], type number),
    #"ln(RR)-1.96*SQRT" = Table.AddColumn(#"ln(RR)+1.96*SQRT", "Subtraction.2",
each [#"ln(RR)"] - [#"1.96*SQRT"], type number),
    #"97.5% CI" = Table.AddColumn(#"ln(RR)-1.96*SQRT", "97.5% CI", each
Number.Exp([Addition]), type number),
    #"2.5% CI" = Table.AddColumn(#"97.5% CI", "2.5% CI", each
Number.Exp([Subtraction.2]), type number),
    #"Removed Columns1" = Table.RemoveColumns(#"2.5% CI",{"1/a", "1/c", "a+b",
"c+d", "1/a+b", "1/c+d", "ln(RR)", "1/a + 1/c", "Subtraction", "Subtraction.1", "Square
Root", "1.96*SQRT", "Addition", "Subtraction.2", "Risk Intervention", "Risk control"}),
```

```
    #"Added Conditional Column" = Table.AddColumn(#"Removed Columns1", "Risk ratio f", each if [Risk ratio] = null then null else if [Risk ratio] >= 0.0001 then [Risk ratio] else null),
    #"Added Conditional Column1" = Table.AddColumn(#"Added Conditional Column", "2.5% CI f", each if [#"2.5% CI"] = null then null else if [#"2.5% CI"] >= 1E-05 then [#"2.5% CI"] else null),
    #"Added Conditional Column2" = Table.AddColumn(#"Added Conditional Column1", "97.5% CI f", each if [#"97.5% CI"] = null then null else if [#"97.5% CI"] >= 1E-05 then [#"97.5% CI"] else null),
    #"Changed Type3" = Table.TransformColumnTypes(#"Added Conditional Column2",{{"Risk ratio f", type number}, {"2.5% CI f", type number}, {"97.5% CI f", type number}})
in
    #"Changed Type3"
```

Overall estimates

```
let
    Source = Excel.Workbook(Web.Contents("data source"), null, true),
    #"Overall estimates_Sheet" = Source{[Item="Overall estimates",Kind="Sheet"]}[Data],
    #"Promoted Headers" = Table.PromoteHeaders(#"Overall estimates_Sheet", [PromoteAllScalars=true]),
    #"Changed Type" = Table.TransformColumnTypes(#"Promoted Headers",{{"Analysis group", Int64.Type}, {"Analysis number", Int64.Type}, {"Analysis name", type text}, {"Analysis group name", type text}, {"Data source", type text}, {"Data source eligibility", type text}, {"Data type", type text}, {"Log-scale data", type logical}, {"Outcome", type text}, {"Intervention grouping", type text}, {"Experimental intervention", type any}, {"Control intervention", type any}, {"Subgroup by", type any}, {"Filter criteria", type any}, {"Experimental group label", type text}, {"Control group label", type text}, {"Statistical method", type text}, {"Effect measure", type text}, {"Unit of effect measure", type any}, {"Analysis model", type text}, {"Heterogeneity estimator", type any}, {"Tau² CI", type logical}, {"Subgroup estimates", type logical}, {"Overall estimates", type logical}, {"Test for subgroup differences", type logical},
```

{"Prediction interval", type logical}, {"Swap event and non-event", type logical}, {"CI method", type text}, {"CI/PI level", type number}, {"Experimental cases", Int64.Type}, {"Experimental N", Int64.Type}, {"Control cases", Int64.Type}, {"Control N", Int64.Type}, {"Mean", type number}, {"CI start", type number}, {"CI end", type number}, {"PI start", type any}, {"PI end", type any}, {"Heterogeneity Tau²", type any}, {"Tau² CI start", type any}, {"Tau² CI end", type any}, {"Heterogeneity Chi²", type number}, {"Heterogeneity df", Int64.Type}, {"Heterogeneity P", type number}, {"Heterogeneity I²", type number}, {"Effect Z", type number}, {"Effect P", type number}, {"Subgroup Chi²", type any}, {"Subgroup df", type any}, {"Subgroup P", type any}, {"Subgroup I²", type any}, {"ID", type text}}),

#"Removed Columns" = Table.RemoveColumns(#"Changed Type",{"PI start", "PI end", "Heterogeneity Tau²", "Tau² CI start", "Tau² CI end", "Heterogeneity Chi²", "Heterogeneity df", "Heterogeneity P", "Effect Z", "Effect P", "Subgroup Chi²", "Subgroup df", "Subgroup P", "ID", "CI method", "Test for subgroup differences", "Prediction interval", "Swap event and non-event", "Tau² CI", "Heterogeneity estimator", "Experimental intervention", "Control intervention"}),

#"Inserted Merged Column" = Table.AddColumn(#"Removed Columns", "GRADE group", each Text.Combine({Text.From([Analysis group], "en-GB"), Text.From([Analysis number], "en-GB")}, "-"), type text),

#"Analyst editable control risk rate (proportion)" = Table.AddColumn(#"Inserted Merged Column", "Control risk rate (proportion)", each if [Outcome] = "a" then 0.2 else null),

#"Changed Type6" = Table.TransformColumnTypes(#"Analyst editable control risk rate (proportion)",{{"Control risk rate (proportion)", type number}}),

#"Adjusted for editable control n calculation" = Table.AddColumn(#"Changed Type6", "Adjusted control N", each [#"Control risk rate (proportion)"] * [Control N], type number),

#"Changed Type8" = Table.TransformColumnTypes(#"Adjusted for editable control n calculation",{{"Adjusted control N", Int64.Type}}),

#"Merged Queries" = Table.NestedJoin(#"Changed Type8", {"GRADE group"}, #"Median baseline rate", {"GRADE number"}, "Median baseline rate", JoinKind.LeftOuter),

```
    #"Expanded Median baseline rate" = Table.ExpandTableColumn(#"Merged
Queries", "Median baseline rate", {"Median control group rate"}, {"Median control
group rate"}),
    #"Adjusted for median control group rate" = Table.AddColumn(#"Expanded
Median baseline rate", "Median control n", each [Median control group rate] * [Control
N], type number),
    #"Changed Type7" = Table.TransformColumnTypes(#"Adjusted for median control
group rate",{{"Median control n", Int64.Type}}),
    #"Renamed Columns4" = Table.RenameColumns(#"Changed Type7",{{"Control
N", "True Control N"}}),
    #"Analyst editable choice of control group N" = Table.AddColumn(#"Renamed
Columns4", "Control N", each if [Outcome] = "a" then [Adjusted control N] else if
[Outcome] = "b" then [Median control n] else if [Outcome] = "c" then [True Control N]
else [True Control N]),
    #"Calculating non-events 1" = Table.AddColumn(#"Analyst editable choice of
control group N", "Non-events 1", each [Experimental N] - [Experimental cases],
Int64.Type),
    #"Calculating non-events 2" = Table.AddColumn(#"Calculating non-events 1",
"Non-events 2", each [Control N] - [Control cases], Int64.Type),
    #"Calculating control rate" = Table.AddColumn(#"Calculating non-events 2",
"Baseline control rate", each ([Control cases] / [Control N]) * 1000, Int64.Type),
    #"Changed Type9" = Table.TransformColumnTypes(#"Calculating control
rate",{{"Baseline control rate", Int64.Type}}),
    #"per 1000 people" = Table.AddColumn(#"Changed Type9", "per 1000 people",
each " per 1000 people"),
    #"Making control group rate" =
Table.CombineColumns(Table.TransformColumnTypes(#"per 1000 people",
{{"Baseline control rate", type text}}, "en-GB"),{"Baseline control rate", "per 1000
people"},Combiner.CombineTextByDelimiter("", QuoteStyle.None),"Control group
rate"),
    #"Replaced Value" = Table.ReplaceValue(#"Making control group rate"," per 1000
people","Not applicable",Replacer.ReplaceValue,{"Control group rate"}),
    #"Calculating risk1" = Table.AddColumn(#"Replaced Value", "Risk1", each
[Experimental cases] / ([Experimental cases] + [Experimental N]), type number),
```

```
    #"Calculating risk2" = Table.AddColumn(#"Calculating risk1", "Risk2", each
[Control cases] / ([Control cases] + [Control N]), type number),
    #"Renamed Columns" = Table.RenameColumns(#"Calculating
risk2",{{"Experimental cases", "Number of events 1"}, {"Experimental N", "n1"},
{"Control cases", "Number of events 2"}, {"Control N", "n2"}}),
    #"Calculating risk difference" = Table.AddColumn(#"Renamed Columns", "Risk
difference", each [Risk1] - [Risk2], type number),
    #"Inserted Multiplication1" = Table.AddColumn(#"Calculating risk difference",
"Event1xNonevent1", each [Number of events 1] * [#"Non-events 1"], Int64.Type),
    #"Inserted Multiplication2" = Table.AddColumn(#"Inserted Multiplication1",
"EventxNonevent2", each [Number of events 2] * [#"Non-events 2"], Int64.Type),
    #"Inserted Cube" = Table.AddColumn(#"Inserted Multiplication2", "n1^3", each
Number.Power([#"Non-events 1"], 3), Int64.Type),
    #"Inserted Cube1" = Table.AddColumn(#"Inserted Cube", "n2^3", each
Number.Power([n2], 3), Int64.Type),
    #"Inserted Division3" = Table.AddColumn(#"Inserted Cube1",
"eventxnonevent1/n1^3", each [Event1xNonevent1] / [#"n1^3"], type number),
    #"Inserted Division4" = Table.AddColumn(#"Inserted Division3", "Division", each
[EventxNonevent2] / [#"n2^3"], type number),
    #"Renamed Columns1" = Table.RenameColumns(#"Inserted
Division4",{{"Division", "eventxnonevent2/n2^3"}}),
    #"Inserted Addition2" = Table.AddColumn(#"Renamed Columns1",
"eventxnonevent1/n1^3 + eventxnonevent2/n2^3", each [#"eventxnonevent1/n1^3"] +
[#"eventxnonevent2/n2^3"], type number),
    #"Inserted Square Root1" = Table.AddColumn(#"Inserted Addition2", "SE(RD)",
each Number.Sqrt([#"eventxnonevent1/n1^3 + eventxnonevent2/n2^3"]), type
number),
    #"Inserted Multiplication3" = Table.AddColumn(#"Inserted Square Root1",
"1.96*SE", each [#"SE(RD)"] * 1.96, type number),
    #"Inserted Subtraction4" = Table.AddColumn(#"Inserted Multiplication3", "RD 2.5%
CI", each [Risk difference] - [#"1.96*SE"], type number),
    #"Inserted Addition3" = Table.AddColumn(#"Inserted Subtraction4", "RD 97.5%
CI", each [Risk difference] + [#"1.96*SE"], type number),
```

```
    #"Removed Columns3" = Table.RemoveColumns(#"Inserted
Addition3",{"Event1xNonevent1", "EventxNonevent2", "n1^3", "n2^3",
"eventxnonevent1/n1^3", "eventxnonevent2/n2^3", "eventxnonevent1/n1^3 +
eventxnonevent2/n2^3", "SE(RD)", "1.96*SE", "Non-events 1", "Non-events 2",
"Risk1"}),
    #"Calculating events/1000 people" = Table.AddColumn(#"Removed Columns3",
"Events/1000 RD", each [Risk difference] * 1000, type number),
    #"Calculating 2.5% CI" = Table.AddColumn(#"Calculating events/1000 people",
"Events/1000 2.5% CI RD", each [#"RD 2.5% CI"] * 1000, type number),
    #"Calculating 97.5% CI" = Table.AddColumn(#"Calculating 2.5% CI", "Events/1000
97.5% CI RD", each [#"RD 97.5% CI"] * 1000, type number),
    #"Calculating events/1000 people mean" = Table.AddColumn(#"Calculating 97.5%
CI", "Events/1000 mean", each [Mean] * 1000, type number),
    #"Calculating 2.5% CI mean" = Table.AddColumn(#"Calculating events/1000
people mean", "Events/1000 2.5% CI mean", each [#"CI start"] * 1000, type number),
    #"Calculating 97.5% CI mean" = Table.AddColumn(#"Calculating 2.5% CI mean",
"Events/1000 97.5% CI mean", each [#"CI end"] * 1000, type number),
    #"Changed Type1" = Table.TransformColumnTypes(#"Calculating 97.5% CI
mean",{{"Events/1000 97.5% CI RD", Int64.Type}, {"Events/1000 2.5% CI RD",
Int64.Type}, {"Events/1000 RD", Int64.Type}, {"Events/1000 97.5% CI mean",
Int64.Type}, {"Events/1000 2.5% CI mean", Int64.Type}, {"Events/1000 mean",
Int64.Type}}),
    #"Analyst choice events/1000 method" = Table.AddColumn(#"Changed Type1",
"Events/1000", each if [Effect measure] = "Risk Difference" then [#"Events/1000
mean"] else if [Analysis name] = "Acute myocardial infarction at end of trial" then
[#"Events/1000 mean"] else [#"Events/1000 RD"]),
    #"Analyst choice 2.5% CI" = Table.AddColumn(#"Analyst choice events/1000
method", "Events/1000 2.5% CI", each if [Effect measure] = "Risk Difference" then
[#"Events/1000 2.5% CI mean"] else if [Analysis name] = "Acute myocardial infarction
at end of study" then [#"Events/1000 2.5% CI mean"] else [#"Events/1000 2.5% CI
RD"]),
    #"Analyst choice 97.5% CI" = Table.AddColumn(#"Analyst choice 2.5% CI",
"Events/1000 97.5% CI", each if [Effect measure] = "Risk Difference" then
[#"Events/1000 97.5% CI mean"] else if [Analysis name] = "Acute myocardial
```

```
infarction at end of study" then [#"Events/1000 97.5% CI mean"] else [#"Events/1000 97.5% CI RD"]),
    #"Extracting RD direction of effect" = Table.AddColumn(#"Analyst choice 97.5% CI", "Sign RD", each Number.Sign([#"Events/1000"]), Int64.Type),
    #"Extracting 2.5% CI direction of effect" = Table.AddColumn(#"Extracting RD direction of effect", "Sign RD 2.5%", each Number.Sign([#"Events/1000 2.5% CI"]), Int64.Type),
    #"Extracting 97.5% CI direction of effect" = Table.AddColumn(#"Extracting 2.5% CI direction of effect", "Sign RD 97.5%", each Number.Sign([#"Events/1000 97.5% CI"]), Int64.Type),
    #"Determining words after RD" = Table.AddColumn(#"Extracting 97.5% CI direction of effect", "RD wording", each if [Sign RD] = 1 then " more events, " else if [Sign RD] = -1 then " fewer events, " else if [Sign RD] = 0 then " fewer events, " else if [Sign RD] = null then null else "?"),
    #"Determining words after RD 2.5% CI" = Table.AddColumn(#"Determining words after RD", "RD 2.5% CI wording", each if [#"Sign RD 2.5%"] = 1 then " more to " else if [#"Sign RD 2.5%"] = -1 then " fewer to " else if [#"Sign RD 2.5%"] = 0 then " fewer to " else if [#"RD 2.5% CI"] = null then null else "?"),
    #"Determining words after RD 97.5% CI" = Table.AddColumn(#"Determining words after RD 2.5% CI", "RD 97.5% CI wording", each if [#"Sign RD 97.5%"] = 1 then " more" else if [#"Sign RD 97.5%"] = -1 then " fewer" else if [#"Sign RD 97.5%"] = 0 then " fewer" else if [#"Sign RD 97.5%"] = null then null else "?"),
    #"Maintain RD direction" = Table.DuplicateColumn(#"Determining words after RD 97.5% CI", "Events/1000", "Events/1000 direction"),
    #"Maintain RD 2.5% direction" = Table.DuplicateColumn(#"Maintain RD direction", "Events/1000 2.5% CI", "Events/1000 2.5% CI direction"),
    #"Maintain RD 97.5% direction" = Table.DuplicateColumn(#"Maintain RD 2.5% direction", "Events/1000 97.5% CI", "Events/1000 97.5% CI direction"),
    #"Calculated Absolute Value" = Table.TransformColumns(#"Maintain RD 97.5% direction",{{"Events/1000", Number.Abs, Int64.Type}, {"Events/1000 2.5% CI", Number.Abs, Int64.Type}, {"Events/1000 97.5% CI", Number.Abs, Int64.Type}}),
    #"Combine for written risk difference value" = Table.AddColumn(#"Calculated Absolute Value", "Risk difference effect", each Text.Combine({Text.From([#"Events/1000"], "en-GB"), [RD wording],
```

```
Text.From([#"Events/1000 2.5% CI"], "en-GB"), [#"RD 2.5% CI wording"],
Text.From([#"Events/1000 97.5% CI"], "en-GB"), [#"RD 97.5% CI wording"]}, ""), type
text),
    #"Added Conditional Column" = Table.AddColumn(#"Combine for written risk
difference value", "Absolute effect", each if [Risk difference effect] <> "" then [Risk
difference effect] else if [Effect measure] = "Risk Ratio" then "?" else if [Effect
measure] = "Odds Ratio" then "?" else "Not applicable"),
    #"Multiplied Column" = Table.TransformColumns(#"Added Conditional Column",
{{"Risk2", each _ * 1000, type number}}),
    #"Renamed Columns2" = Table.RenameColumns(#"Multiplied Column",{{"Risk2",
"Number of events in control group/1000 people"}}),
    #"Changed Type2" = Table.TransformColumnTypes(#"Renamed
Columns2",{{"Number of events in control group/1000 people", Int64.Type}}),
    #"Removed Columns2" = Table.RemoveColumns(#"Changed Type2",{"Sign RD",
"Sign RD 2.5%", "Sign RD 97.5%"}),
    #"Inserted Addition" = Table.AddColumn(#"Removed Columns2", "Total events
after treatment", each [#"Number of events in control group/1000 people"] +
[#"Events/1000 direction"], Int64.Type),
    #"Determining smallest number of events/1000" = Table.AddColumn(#"Inserted
Addition", "Smallest number of events/1000", each if [Total events after treatment] =
null then null else if [Total events after treatment] < [#"Number of events in control
group/1000 people"] then [Total events after treatment] else [#"Number of events in
control group/1000 people"]),
    #"Changed Type4" = Table.TransformColumnTypes(#"Determining smallest
number of events/1000",{{"Smallest number of events/1000", Int64.Type}}),
    #"Added Custom" = Table.AddColumn(#"Changed Type4", "1000", each "1000"),
    #"Inserted Subtraction" = Table.AddColumn(#"Added Custom", "Additional events
from treatment", each [Total events after treatment] - [#"Smallest number of
events/1000"], type number),
    #"Inserted Subtraction1" = Table.AddColumn(#"Inserted Subtraction", "Events
reduced from treatment", each [#"Number of events in control group/1000 people"] -
[#"Smallest number of events/1000"], type number),
    #"Changed Type3" = Table.TransformColumnTypes(#"Inserted
Subtraction1",{{"1000", Int64.Type}, {"Smallest number of events/1000", Int64.Type},
```

{"Additional events from treatment", Int64.Type}, {"Events reduced from treatment", Int64.Type}}),
#"Determining largest number of events/1000" = Table.AddColumn(#"Changed Type3", "Largest number of events/1000", each if [#"Number of events in control group/1000 people"] = null then null else if [#"Number of events in control group/1000 people"] >= [Total events after treatment] then [#"Number of events in control group/1000 people"] else [Total events after treatment]),
#"Changed Type5" = Table.TransformColumnTypes(#"Determining largest number of events/1000",{{"Largest number of events/1000", Int64.Type}}),
#"Calculating people without event" = Table.AddColumn(#"Changed Type5", "People without event", each [1000] - [#"Largest number of events/1000"], Int64.Type),
#"Renamed Columns3" = Table.RenameColumns(#"Calculating people without event",{{"Smallest number of events/1000", "People with events"}, {"Events reduced from treatment", "Fewer events after treatment"}}),
#"Added Custom1" = Table.AddColumn(#"Renamed Columns3", "|", each "|"),
#"Filtered Rows" = Table.SelectRows(#"Added Custom1", each ([Analysis group] = 1)),
#"Added Custom2" = Table.AddColumn(#"Filtered Rows", "(", each "("),
#"Added Custom3" = Table.AddColumn(#"Added Custom2", ")", each ")"),
#"Rounded Off" = Table.TransformColumns(#"Added Custom3",{{"Mean", each Number.Round(_, 5), type number}, {"CI start", each Number.Round(_, 5), type number}, {"CI end", each Number.Round(_, 5), type number}}),
#"Inserted Merged Column1" = Table.AddColumn(#"Rounded Off", "Effect size", each Text.Combine({Text.From([Mean], "en-GB"), [#"("]}, " "), type text),
#"Inserted Merged Column2" = Table.AddColumn(#"Inserted Merged Column1", "Effect Size.1", each Text.Combine({[Effect size], Text.From([CI start], "en-GB")}, ""), type text),
#"Inserted Merged Column3" = Table.AddColumn(#"Inserted Merged Column2", "Effect size.2", each Text.Combine({[Effect Size.1], Text.From([CI end], "en-GB")}, ", "), type text),
#"Inserted Merged Column4" = Table.AddColumn(#"Inserted Merged Column3", "Effect size.3", each Text.Combine({[Effect size.2], [#")"]}, ""), type text),

```
    #"Removed Columns1" = Table.RemoveColumns(#"Inserted Merged
Column4",{"(", ")", "Effect size", "Effect Size.1", "Effect size.2"}),
    #"Renamed Columns5" = Table.RenameColumns(#"Removed Columns1",{{"Effect
size.3", "Effect size"}}),
    #"Risk intervention" = Table.AddColumn(#"Renamed Columns5", "Risk
Intervention", each [Number of events 1] / [n1], type number),
    #"Risk control" = Table.AddColumn(#"Risk intervention", "Risk control", each
[Number of events 2] / [True Control N], type number),
    #"Risk ratio" = Table.AddColumn(#"Risk control", "Risk ratio", each [Risk
Intervention] / [Risk control], type number),
    #"1/a" = Table.AddColumn(#"Risk ratio", "1/a", each Number.Power([Number of
events 1], -1), Int64.Type),
    #"1/c" = Table.AddColumn(#"1/a", "1/c", each Number.Power([Number of events
2], -1), Int64.Type),
    #"a+b" = Table.AddColumn(#"1/c", "a+b", each [Number of events 1] + [n1],
Int64.Type),
    #"c+d" = Table.AddColumn(#"a+b", "c+d", each [Number of events 2] + [True
Control N], Int64.Type),
    #"1/a+b" = Table.AddColumn(#"c+d", "1/a+b", each Number.Power([#"a+b"], -1),
type number),
    #"1/c+d" = Table.AddColumn(#"1/a+b", "1/c+d", each Number.Power([#"c+d"], -1),
type number),
    #"Inserted Natural Logarithm" = Table.AddColumn(#"1/c+d", "ln(RR)", each
Number.Ln([Risk ratio]), type number),
    #"Inserted Addition4" = Table.AddColumn(#"Inserted Natural Logarithm", "1/a +
1/c", each [#"1/a"] + [#"1/c"], Int64.Type),
    #"Inserted Subtraction2" = Table.AddColumn(#"Inserted Addition4", "Subtraction",
each [#"1/a + 1/c"] - [#"1/a+b"], type number),
    #"Inserted Subtraction3" = Table.AddColumn(#"Inserted Subtraction2",
"Subtraction.1", each [Subtraction] - [#"1/c+d"], type number),
    #"Inserted Square Root" = Table.AddColumn(#"Inserted Subtraction3", "Square
Root", each Number.Sqrt([Subtraction.1]), type number),
    #"1.96*SQRT" = Table.AddColumn(#"Inserted Square Root", "1.96*SQRT", each
[Square Root] * 1.96, type number),
```

```
    #"ln(RR)+1.96*SQRT" = Table.AddColumn(#"1.96*SQRT", "Addition", each
[#"ln(RR)"] + [#"1.96*SQRT"], type number),
    #"ln(RR)-1.96*SQRT" = Table.AddColumn(#"ln(RR)+1.96*SQRT", "Subtraction.2",
each [#"ln(RR)"] - [#"1.96*SQRT"], type number),
    #"97.5% CI" = Table.AddColumn(#"ln(RR)-1.96*SQRT", "97.5% CI", each
Number.Exp([Addition]), type number),
    #"2.5% CI" = Table.AddColumn(#"97.5% CI", "2.5% CI", each
Number.Exp([Subtraction.2]), type number),
    #"Removed Columns4" = Table.RemoveColumns(#"2.5% CI",{"1/a", "1/c", "a+b",
"c+d", "1/a+b", "1/c+d", "ln(RR)", "1/a + 1/c", "Subtraction", "Subtraction.1", "Square
Root", "1.96*SQRT", "Addition", "Subtraction.2", "Risk Intervention", "Risk control"}),
    #"Added Conditional Column1" = Table.AddColumn(#"Removed Columns4", "Risk
ratio f", each if [Risk ratio] = null then null else if [Risk ratio] >= 0.0001 then [Risk
ratio] else null),
    #"Added Conditional Column2" = Table.AddColumn(#"Added Conditional
Column1", "2.5% CI f", each if [#"2.5% CI"] = null then null else if [#"2.5% CI"] >= 1E-
05 then [#"2.5% CI"] else null),
    #"Added Conditional Column3" = Table.AddColumn(#"Added Conditional
Column2", "97.5% CI f", each if [#"97.5% CI"] = null then null else if [#"97.5% CI"] >=
1E-05 then [#"97.5% CI"] else null),
    #"Changed Type0" = Table.TransformColumnTypes(#"Added Conditional
Column3",{{"Risk ratio f", type number}, {"2.5% CI f", type number}, {"97.5% CI f",
type number}}),
    #"Added Conditional Column4" = Table.AddColumn(#"Changed Type0", "Risk
differencef", each if [Effect measure] = "Risk Difference" then [Mean] else [Risk
difference]),
    #"Added Conditional Column5" = Table.AddColumn(#"Added Conditional
Column4", "RD 2.5% CI f", each if [Effect measure] = "Risk Difference" then [CI start]
else [#"RD 2.5% CI"]),
    #"Added Conditional Column6" = Table.AddColumn(#"Added Conditional
Column5", "RD 97.5% CI f", each if [Effect measure] = "Risk Difference" then [CI end]
else [#"RD 97.5% CI"]),
```

```
    #"Changed Type10" = Table.TransformColumnTypes(#"Added Conditional
Column6",{{"Risk differencef", type number}, {"RD 2.5% CI f", type number}, {"RD
97.5% CI f", type number}})
in
    #"Changed Type10"
```

Risk of bias

```
let
    Source = Excel.Workbook(Web.Contents("source"), null, true),
    #"Data rows_Sheet" = Source{[Item="Data rows",Kind="Sheet"]}[Data],
    #"Promoted Headers" = Table.PromoteHeaders(#"Data rows_Sheet",
[PromoteAllScalars=true]),
    #"Changed Type" = Table.TransformColumnTypes(#"Promoted
Headers",{{"Analysis group", Int64.Type}, {"Analysis number", Int64.Type}, {"Analysis
name", type text}, {"Subgroup", type any}, {"Applicability", type text}, {"Study", type
text}, {"Study year", Int64.Type}, {"GIV Mean", type number}, {"GIV SE", type
number}, {"Experimental mean", type number}, {"Experimental SD", type number},
{"Experimental cases", Int64.Type}, {"Experimental N", Int64.Type}, {"Control mean",
type number}, {"Control SD", type number}, {"Control cases", Int64.Type}, {"Control
N", Int64.Type}, {"O-E", type any}, {"Variance", type any}, {"Weight", type number},
{"Mean", type number}, {"CI start", type number}, {"CI end", type number},
{"Footnotes", type any}}),
    #"Splitting by "" at "" to separate follow up time" = Table.SplitColumn(#"Changed
Type", "Analysis name", Splitter.SplitTextByDelimiter(" at ", QuoteStyle.Csv),
{"Analysis name.1", "Analysis name.2"}),
    #"Changed Type1" = Table.TransformColumnTypes(#"Splitting by "" at "" to
separate follow up time",{{"Analysis name.1", type text}, {"Analysis name.2", type
text}}),
    #"Renamed Columns" = Table.RenameColumns(#"Changed Type1",{{"Analysis
name.2", "Follow up time"}}),
    #"Splitting by "" ("" to separate scale information" = Table.SplitColumn(#"Renamed
Columns", "Analysis name.1", Splitter.SplitTextByDelimiter(" (", QuoteStyle.Csv),
{"Analysis name.1.1", "Analysis name.1.2"}),
```

```
    #"Changed Type2" = Table.TransformColumnTypes(#"Splitting by "" ("" to
separate scale information",{{"Analysis name.1.1", type text}, {"Analysis name.1.2",
type text}}),
    #"Renamed Columns1" = Table.RenameColumns(#"Changed Type2",{{"Analysis
name.1.2", "Scale"}, {"Analysis name.1.1", "Protocol Outcome"}}),
    #"Removing closing bracket" = Table.ReplaceValue(#"Renamed
Columns1",")","",Replacer.ReplaceText,{"Scale"}),
    #"Making short outcome" = Table.AddColumn(#"Removing closing bracket",
"Outcome short", each if [Protocol Outcome] = "Mortality" then "Mort" else if [Protocol
Outcome] = "Physical dependency" then "Pdep" else if [Protocol Outcome] = "Falls"
then "Falls" else if [Protocol Outcome] = "Readmissions to hospital" then "Read" else
if [Protocol Outcome] = "Person/participant generic health-related quality of life" then
"HQOL" else if [Protocol Outcome] = "Activities of daily living" then "ADL" else if
[Protocol Outcome] = "Extended activities of daily living" then "EADL" else if [Protocol
Outcome] = "Length of hospital stay" then "LOS" else if [Protocol Outcome] =
"Caregiver strain index" then "CSI" else if [Protocol Outcome] = "Psychological
distress/mood" then "PD" else if [Protocol Outcome] = "Carer generic health-related
quality of life" then "CHQOL" else if [Protocol Outcome] = "Stroke-specific Patient-
Reported Outcome Measures" then "SQOL" else "?"),
    #"Merged Queries" = Table.NestedJoin(#"Making short outcome", {"Study"},
BiasInd, {"Study"}, "Critical appraisal no edits", JoinKind.LeftOuter),
    #"Expanded Critical appraisal no edits" = Table.ExpandTableColumn(#"Merged
Queries", "Critical appraisal no edits", {"Overall risk of bias", "Overall directness",
"Short population", "Short comparison"}, {"Overall risk of bias", "Overall directness",
"Short population", "Short comparison"}),
    #"Removed Duplicates" = Table.Distinct(#"Expanded Critical appraisal no edits"),
    #"Removed Columns" = Table.RemoveColumns(#"Removed
Duplicates",{"Footnotes", "GIV Mean", "GIV SE", "Experimental mean", "Experimental
SD", "Experimental cases", "Experimental N", "Control mean", "Control SD", "Control
cases", "Control N", "O-E", "Variance", "Mean", "CI start", "CI end"}),
    #"Timepoint = 0" = Table.AddColumn(#"Removed Columns", "Timepoint
(months)", each 0),
    #"Reordered Columns" = Table.ReorderColumns(#"Timepoint = 0",{"Analysis
group", "Analysis number", "Protocol Outcome", "Scale", "Outcome short", "Short
```

population", "Short comparison", "Timepoint (months)", "Follow up time", "Subgroup", "Applicability", "Study", "Study year", "Weight", "Overall risk of bias", "Overall directness"}),
#"Removed Columns1" = Table.RemoveColumns(#"Reordered Columns",{"Follow up time"}),
#"Reordered Columns1" = Table.ReorderColumns(#"Removed Columns1",{"Analysis group", "Analysis number", "Study", "Protocol Outcome", "Scale", "Outcome short", "Short population", "Short comparison", "Timepoint (months)", "Subgroup", "Applicability", "Study year", "Weight", "Overall risk of bias", "Overall directness"}),
#"Removed Columns2" = Table.RemoveColumns(#"Reordered Columns1",{"Study year"}),
#"Changing RoB from text to numbers" = Table.AddColumn(#"Removed Columns2", "Risk of bias number", each if [Overall risk of bias] = "Low" then 0 else if [Overall risk of bias] = "Some concerns" then 1 else if [Overall risk of bias] = "High" then 2 else "?"),
#"Removed Columns3" = Table.RemoveColumns(#"Changing RoB from text to numbers",{"Study", "Short population", "Short comparison", "Subgroup", "Applicability", "Timepoint (months)"}),
#"Calculating weighted RoB" = Table.AddColumn(#"Removed Columns3", "Weighted RoB", each [Risk of bias number] * [Weight], type number),
#"Dividing by 100" = Table.TransformColumns(#"Calculating weighted RoB", {{"Weighted RoB", each _ / 100, type number}}),
#"Removed Columns4" = Table.RemoveColumns(#"Dividing by 100",{"Weight", "Overall risk of bias", "Overall directness", "Risk of bias number"}),
#"Duplicated Column" = Table.DuplicateColumn(#"Removed Columns4", "Protocol Outcome", "Protocol Outcome - Copy"),
#"Pivoted Column to calculate combined RoB" = Table.Pivot(#"Duplicated Column", List.Distinct(#"Duplicated Column"[#"Protocol Outcome - Copy"]), "Protocol Outcome - Copy", "Weighted RoB", List.Sum),
#"Merged Columns" = Table.CombineColumns(Table.TransformColumnTypes(#"Pivoted Column to calculate combined RoB", {{"Proportion of patients requiring transfusion", type text}, {"Thrombotic complications", type text}, {"Duplicate outcome proportion of people

requiring transfusion", type text}, {"Serious adverse events", type text}, {"All-cause mortality", type text}, {"Number of units of allogenic blood transfused", type text}, {"Infection", type text}, {"Length of stay", type text}, {"Acute myocardial infarction", type text}, {"Volume of allogenic blood transfused", type text}, {"Total blood loss volume", type text}}, "en-GB"),{"Proportion of patients requiring transfusion", "Thrombotic complications", "Duplicate outcome proportion of people requiring transfusion", "Serious adverse events", "All-cause mortality", "Number of units of allogenic blood transfused", "Infection", "Length of stay", "Acute myocardial infarction", "Volume of allogenic blood transfused", "Total blood loss volume"},Combiner.CombineTextByDelimiter("", QuoteStyle.None),"Summed weighted RoB Number"),

#"Changed Type3" = Table.TransformColumnTypes(#"Merged Columns",{{"Summed weighted RoB Number", type number}}),

#"Calculation of RoB" = Table.AddColumn(#"Changed Type3", "Outcome RoB", each if [Summed weighted RoB Number] < 0 then "?" else if [Summed weighted RoB Number] <= 0.66 then "Not serious" else if [Summed weighted RoB Number] <= 1.32 then "Serious" else if [Summed weighted RoB Number] <= 2 then "Very serious" else "?"),

#"Analyst manual edit of RoB" = Table.AddColumn(#"Calculation of RoB", "Edited outcome RoB", each if [Protocol Outcome] = "Falls" then "Serious" else null),

#"Risk of Bias final determination" = Table.AddColumn(#"Analyst manual edit of RoB", "RoB", each if [Edited outcome RoB] <> null then [Edited outcome RoB] else [Outcome RoB]),

#"Risk of bias reason determination" = Table.AddColumn(#"Risk of Bias final determination", "RoB reason", each if [RoB] = "Serious" then "Risk of bias: Downgraded once. Serious risk of bias in the evidence contributing to the outcomes. More than 50% of the weight of the evidence came from studies at moderate or high risk of bias as per RoB2" else if [RoB] = "Very serious" then "Risk of bias: Downgraded twice. Very serious risk of bias in the evidence contributing to the outcomes. More than 50% of the weight of the evidence came from studies at high risk of bias as per RoB2" else null),

#"Inserted Merged Column" = Table.AddColumn(#"Risk of bias reason determination", "GRADE code", each Text.Combine({Text.From([Analysis group], "en-GB"), Text.From([Analysis number], "en-GB")}, "-"), type text),

```
    #"Removed Columns5" = Table.RemoveColumns(#"Inserted Merged
Column",{"Protocol Outcome", "Scale", "Outcome short", "Summed weighted RoB
Number", "Outcome RoB", "Edited outcome RoB"}),
    #"Reordered Columns2" = Table.ReorderColumns(#"Removed
Columns5",{"Analysis group", "Analysis number", "GRADE code", "RoB", "RoB
reason"})
in
    #"Reordered Columns2"
```

Indirectness

```
let
    Source = Excel.Workbook(Web.Contents("source"), null, true),
    #"Data rows_Sheet" = Source{[Item="Data rows",Kind="Sheet"]}[Data],
    #"Promoted Headers" = Table.PromoteHeaders(#"Data rows_Sheet",
[PromoteAllScalars=true]),
    #"Changed Type" = Table.TransformColumnTypes(#"Promoted
Headers",{{"Analysis group", Int64.Type}, {"Analysis number", Int64.Type}, {"Analysis
name", type text}, {"Subgroup", type any}, {"Applicability", type text}, {"Study", type
text}, {"Study year", Int64.Type}, {"GIV Mean", type number}, {"GIV SE", type
number}, {"Experimental mean", type number}, {"Experimental SD", type number},
{"Experimental cases", Int64.Type}, {"Experimental N", Int64.Type}, {"Control mean",
type number}, {"Control SD", type number}, {"Control cases", Int64.Type}, {"Control
N", Int64.Type}, {"O-E", type any}, {"Variance", type any}, {"Weight", type number},
{"Mean", type number}, {"CI start", type number}, {"CI end", type number},
{"Footnotes", type any}}),
    #"Splitting by "" at "" to separate follow up time" = Table.SplitColumn(#"Changed
Type", "Analysis name", Splitter.SplitTextByDelimiter(" at ", QuoteStyle.Csv),
{"Analysis name.1", "Analysis name.2"}),
    #"Changed Type1" = Table.TransformColumnTypes(#"Splitting by "" at "" to
separate follow up time",{{"Analysis name.1", type text}, {"Analysis name.2", type
text}}),
    #"Renamed Columns" = Table.RenameColumns(#"Changed Type1",{{"Analysis
name.2", "Follow up time"}}),
```

```
    #"Splitting by "" ("" to separate scale information" = Table.SplitColumn(#"Renamed Columns", "Analysis name.1", Splitter.SplitTextByDelimiter(" (", QuoteStyle.Csv), {"Analysis name.1.1", "Analysis name.1.2"}),
    #"Changed Type2" = Table.TransformColumnTypes(#"Splitting by "" ("" to separate scale information",{{"Analysis name.1.1", type text}, {"Analysis name.1.2", type text}}),
    #"Renamed Columns1" = Table.RenameColumns(#"Changed Type2",{{"Analysis name.1.2", "Scale"}, {"Analysis name.1.1", "Protocol Outcome"}}),
    #"Removing closing bracket" = Table.ReplaceValue(#"Renamed Columns1",")","",Replacer.ReplaceText,{"Scale"}),
    #"Making short outcome" = Table.AddColumn(#"Removing closing bracket", "Outcome short", each if [Protocol Outcome] = "Mortality" then "Mort" else if [Protocol Outcome] = "Physical dependency" then "Pdep" else if [Protocol Outcome] = "Falls" then "Falls" else if [Protocol Outcome] = "Readmissions to hospital" then "Read" else if [Protocol Outcome] = "Person/participant generic health-related quality of life" then "HQOL" else if [Protocol Outcome] = "Activities of daily living" then "ADL" else if [Protocol Outcome] = "Extended activities of daily living" then "EADL" else if [Protocol Outcome] = "Length of hospital stay" then "LOS" else if [Protocol Outcome] = "Caregiver strain index" then "CSI" else if [Protocol Outcome] = "Psychological distress/mood" then "PD" else if [Protocol Outcome] = "Carer generic health-related quality of life" then "CHQOL" else if [Protocol Outcome] = "Stroke-specific Patient-Reported Outcome Measures" then "SQOL" else "?"),
    #"Merged Queries" = Table.NestedJoin(#"Making short outcome", {"Study"}, BiasInd, {"Study"}, "Critical appraisal no edits", JoinKind.LeftOuter),
    #"Expanded Critical appraisal no edits" = Table.ExpandTableColumn(#"Merged Queries", "Critical appraisal no edits", {"Overall risk of bias", "Overall directness", "Short population", "Short comparison"}, {"Overall risk of bias", "Overall directness", "Short population", "Short comparison"}),
    #"Removed Duplicates" = Table.Distinct(#"Expanded Critical appraisal no edits"),
    #"Removed Columns" = Table.RemoveColumns(#"Removed Duplicates",{"Footnotes", "GIV Mean", "GIV SE", "Experimental mean", "Experimental SD", "Experimental cases", "Experimental N", "Control mean", "Control SD", "Control cases", "Control N", "O-E", "Variance", "Mean", "CI start", "CI end"}),
```

#"Timepoint = 0" = Table.AddColumn(#"Removed Columns", "Timepoint (months)", each 0),
#"Removed Columns1" = Table.RemoveColumns(#"Timepoint = 0",{"Follow up time"}),
#"Reordered Columns1" = Table.ReorderColumns(#"Removed Columns1",{"Analysis group", "Analysis number", "Study", "Protocol Outcome", "Scale", "Outcome short", "Short population", "Short comparison", "Timepoint (months)", "Subgroup", "Applicability", "Study year", "Weight", "Overall risk of bias", "Overall directness"}),
#"Removed Columns2" = Table.RemoveColumns(#"Reordered Columns1",{"Study year"}),
#"Changing directness from text to numbers" = Table.AddColumn(#"Removed Columns2", "Directness number", each if [Overall directness] = "Directly applicable" then 0 else if [Overall directness] = "Partially applicable" then 1 else if [Overall directness] = "Indirectly applicable" then 2 else "?"),
#"Removed Columns3" = Table.RemoveColumns(#"Changing directness from text to numbers",{"Study", "Short population", "Short comparison", "Timepoint (months)", "Subgroup", "Applicability"}),
#"Calculating weighted directness" = Table.AddColumn(#"Removed Columns3", "Weighted directness", each [Directness number] * [Weight], type number),
#"Dividing by 100" = Table.TransformColumns(#"Calculating weighted directness", {{"Weighted directness", each _ / 100, type number}}),
#"Removed Columns4" = Table.RemoveColumns(#"Dividing by 100",{"Weight", "Overall risk of bias", "Overall directness", "Directness number"}),
#"Duplicated Column" = Table.DuplicateColumn(#"Removed Columns4", "Protocol Outcome", "Protocol Outcome - Copy"),
#"Pivoted Column to calculate combined directness" = Table.Pivot(#"Duplicated Column", List.Distinct(#"Duplicated Column"[#"Protocol Outcome - Copy"]), "Protocol Outcome - Copy", "Weighted directness", List.Sum),
#"Merged Columns" = Table.CombineColumns(Table.TransformColumnTypes(#"Pivoted Column to calculate combined directness", {{"Proportion of patients requiring transfusion", type text}, {"Thrombotic complications", type text}, {"Duplicate outcome proportion of people requiring transfusion", type text}, {"Serious adverse events", type text}, {"All-

cause mortality", type text}, {"Number of units of allogenic blood transfused", type text}, {"Infection", type text}, {"Length of stay", type text}, {"Acute myocardial infarction", type text}, {"Volume of allogenic blood transfused", type text}, {"Total blood loss volume", type text}}, "en-GB"),{"Proportion of patients requiring transfusion", "Thrombotic complications", "Duplicate outcome proportion of people requiring transfusion", "Serious adverse events", "All-cause mortality", "Number of units of allogenic blood transfused", "Infection", "Length of stay", "Acute myocardial infarction", "Volume of allogenic blood transfused", "Total blood loss volume"},Combiner.CombineTextByDelimiter("", QuoteStyle.None),"Summed weighted directness number"),

#"Changed Type3" = Table.TransformColumnTypes(#"Merged Columns",{{"Summed weighted directness number", type number}}),

#"Calculation of directness" = Table.AddColumn(#"Changed Type3", "Outcome directness", each if [Summed weighted directness number] < 0 then "?" else if [Summed weighted directness number] <= 0.66 then "Not serious" else if [Summed weighted directness number] <= 1.32 then "Serious" else if [Summed weighted directness number] <= 2 then "Very serious" else "?"),

#"Analyst manual edit of directness" = Table.AddColumn(#"Calculation of directness", "Edited outcome directness", each if [Protocol Outcome] = "Falls" then null else null),

#"Directness final determination" = Table.AddColumn(#"Analyst manual edit of directness", "Indirectness", each if [Edited outcome directness] <> null then [Edited outcome directness] else [Outcome directness]),

#"Indirectness reason determination" = Table.AddColumn(#"Directness final determination", "Indirectness reason", each if [Protocol Outcome] = "Physical dependency" then "Indirectness: Downgraded once. Serious indirectness due to >50% of overall weighting partially direct or indirect. Outcome indirect due to it including mortality and physical dependency" else null),

#"Inserted Merged Column" = Table.AddColumn(#"Indirectness reason determination", "GRADE code", each Text.Combine({Text.From([Analysis group], "en-GB"), Text.From([Analysis number], "en-GB")}, "-"), type text),

#"Removed Columns5" = Table.RemoveColumns(#"Inserted Merged Column",{"Protocol Outcome", "Scale", "Outcome short", "Summed weighted directness number", "Outcome directness", "Edited outcome directness"}),

```
    #"Reordered Columns2" = Table.ReorderColumns(#"Removed
Columns5",{"Analysis group", "Analysis number", "GRADE code", "Indirectness",
"Indirectness reason"})
in
    #"Reordered Columns2"
```

Imprecision

```
let
    Source = Excel.Workbook(Web.Contents("source"), null, true),
    #"Overall estimates_Sheet" = Source{[Item="Overall
estimates",Kind="Sheet"]}[Data],
    #"Promoted Headers" = Table.PromoteHeaders(#"Overall estimates_Sheet",
[PromoteAllScalars=true]),
    #"Changed Type" = Table.TransformColumnTypes(#"Promoted
Headers",{{"Analysis group", Int64.Type}, {"Analysis number", Int64.Type}, {"Analysis
name", type text}, {"Analysis group name", type text}, {"Data source", type text},
{"Data source eligibility", type text}, {"Data type", type text}, {"Log-scale data", type
logical}, {"Outcome", type text}, {"Intervention grouping", type text}, {"Experimental
intervention", type any}, {"Control intervention", type any}, {"Subgroup by", type any},
{"Filter criteria", type any}, {"Experimental group label", type text}, {"Control group
label", type text}, {"Statistical method", type text}, {"Effect measure", type text}, {"Unit
of effect measure", type any}, {"Analysis model", type text}, {"Heterogeneity
estimator", type any}, {"Tau² CI", type logical}, {"Subgroup estimates", type logical},
{"Overall estimates", type logical}, {"Test for subgroup differences", type logical},
{"Prediction interval", type logical}, {"Swap event and non-event", type logical}, {"CI
method", type text}, {"CI/PI level", type number}, {"Experimental cases", Int64.Type},
{"Experimental N", Int64.Type}, {"Control cases", Int64.Type}, {"Control N",
Int64.Type}, {"Mean", type number}, {"CI start", type number}, {"CI end", type
number}, {"PI start", type any}, {"PI end", type any}, {"Heterogeneity Tau²", type any},
{"Tau² CI start", type any}, {"Tau² CI end", type any}, {"Heterogeneity Chi²", type
number}, {"Heterogeneity df", Int64.Type}, {"Heterogeneity P", type number},
{"Heterogeneity I²", type number}, {"Effect Z", type number}, {"Effect P", type
```

```
number}, {"Subgroup Chi²", type any}, {"Subgroup df", type any}, {"Subgroup P", type any}, {"Subgroup I²", type any}, {"ID", type text}}),
    #"Filtered Rows" = Table.SelectRows(#"Changed Type", each ([Analysis group] <> 5 and [Analysis group] <> 6)),
    #"Removed Other Columns" = Table.SelectColumns(#"Filtered Rows",{"Analysis group", "Analysis number", "Analysis name", "Analysis group name", "Outcome", "Experimental intervention", "Control intervention", "Effect measure", "Experimental cases", "Experimental N", "Control cases", "Control N", "Mean", "CI start", "CI end"}),
    #"Calculating sample size" = Table.AddColumn(#"Removed Other Columns", "Sample size", each [Experimental N] + [Control N], Int64.Type),
    #"Renamed Columns" = Table.RenameColumns(#"Calculating sample size",{{"Experimental N", "n1"}, {"Control N", "n2"}, {"Experimental cases", "Number of events 1"}, {"Control cases", "Number of events 2"}}),
    #"Calculating non-events 1" = Table.AddColumn(#"Renamed Columns", "Non-events 1", each [n1] - [Number of events 1], Int64.Type),
    #"Calculating non-events 2" = Table.AddColumn(#"Calculating non-events 1", "Non-events 2", each [n2] - [Number of events 2], Int64.Type),
    #"Calculating Risk1" = Table.AddColumn(#"Calculating non-events 2", "Risk1", each [Number of events 1] / [n1], type number),
    #"Calculating Risk2" = Table.AddColumn(#"Calculating Risk1", "Risk2", each [Number of events 2] / [n2], type number),
    #"Calculating risk difference" = Table.AddColumn(#"Calculating Risk2", "Risk difference", each [Risk1] - [Risk2], type number),
    #"Inserted Multiplication1" = Table.AddColumn(#"Calculating risk difference", "Event1xNonevent1", each [Number of events 1] * [#"Non-events 1"], Int64.Type),
    #"Inserted Multiplication2" = Table.AddColumn(#"Inserted Multiplication1", "EventxNonevent2", each [Number of events 2] * [#"Non-events 2"], Int64.Type),
    #"Inserted Cube" = Table.AddColumn(#"Inserted Multiplication2", "n1^3", each Number.Power([#"Non-events 1"], 3), Int64.Type),
    #"Inserted Cube1" = Table.AddColumn(#"Inserted Cube", "n2^3", each Number.Power([n2], 3), Int64.Type),
    #"Inserted Division3" = Table.AddColumn(#"Inserted Cube1", "eventxnonevent1/n1^3", each [Event1xNonevent1] / [#"n1^3"], type number),
```

```
    #"Inserted Division4" = Table.AddColumn(#"Inserted Division3", "Division", each [EventxNonevent2] / [#"n2^3"], type number),
    #"Renamed Columns1" = Table.RenameColumns(#"Inserted Division4",{{"Division", "eventxnonevent2/n2^3"}}),
    #"Inserted Addition2" = Table.AddColumn(#"Renamed Columns1", "eventxnonevent1/n1^3 + eventxnonevent2/n2^3", each [#"eventxnonevent1/n1^3"] + [#"eventxnonevent2/n2^3"], type number),
    #"Calculate SE(RD)" = Table.AddColumn(#"Inserted Addition2", "SE(RD)", each Number.Sqrt([#"eventxnonevent1/n1^3 + eventxnonevent2/n2^3"]), type number),
    #"Inserted Multiplication3" = Table.AddColumn(#"Calculate SE(RD)", "1.96*SE", each [#"SE(RD)"] * 1.96, type number),
    #"Calculate RD 2.5% CI" = Table.AddColumn(#"Inserted Multiplication3", "RD 2.5% CI", each [Risk difference] - [#"1.96*SE"], type number),
    #"Calculate RD 97.5% CI" = Table.AddColumn(#"Calculate RD 2.5% CI", "RD 97.5% CI", each [Risk difference] + [#"1.96*SE"], type number),
    #"Calculating events/1000 people" = Table.AddColumn(#"Calculate RD 97.5% CI", "Events/1000", each [Risk difference] * 1000, type number),
    #"Calculating 2.5% CI" = Table.AddColumn(#"Calculating events/1000 people", "Events/1000 2.5% CI", each [#"RD 2.5% CI"] * 1000, type number),
    #"Calculating 97.5% CI" = Table.AddColumn(#"Calculating 2.5% CI", "Events/1000 97.5% CI", each [#"RD 97.5% CI"] * 1000, type number),
    #"Changed Type2" = Table.TransformColumnTypes(#"Calculating 97.5% CI",{{"Events/1000 97.5% CI", Int64.Type}, {"Events/1000 2.5% CI", Int64.Type}, {"Events/1000", Int64.Type}}),
    #"Removed Other Columns1" = Table.SelectColumns(#"Changed Type2",{"Analysis group", "Analysis number", "Analysis name", "Analysis group name", "Outcome", "Experimental intervention", "Control intervention", "Effect measure", "Number of events 1", "n1", "Number of events 2", "n2", "Mean", "CI start", "CI end", "Sample size", "Events/1000", "Events/1000 2.5% CI", "Events/1000 97.5% CI"}),
    #"Subtracting CIs to calculate SE" = Table.AddColumn(#"Removed Other Columns1", "SE", each [CI end] - [CI start], type number),
    #"Dividing by 3.92 to calculate SE" = Table.TransformColumns(#"Subtracting CIs to calculate SE", {{"SE", each _ / 3.92, type number}}),
```

```
    #"Square Root of sample size" = Table.AddColumn(#"Dividing by 3.92 to calculate SE", "Square Root(N)", each Number.Sqrt([Sample size]), type number),
    #"Calculate SD" = Table.AddColumn(#"Square Root of sample size", "SD", each [#"Square Root(N)"] * [SE], type number),
    #"SD squared" = Table.AddColumn(#"Calculate SD", "SD^2", each Number.Power([SD], 2), type number),
    #"Choosing effect for MID precision calc" = Table.AddColumn(#"SD squared", "Events for MIDs", each if [Effect measure] = "Risk Ratio" then [Mean] else if [Effect measure] = "Odds Ratio" then [Mean] else if [Effect measure] = "Risk Difference" then [#"Events/1000"] else [Mean]),
    #"Choosing 2.5% CI for MID precision calc" = Table.AddColumn(#"Choosing effect for MID precision calc", "2.5% CI for MIDs", each if [Effect measure] = "Risk Ratio" then [CI start] else if [Effect measure] = "Odds Ratio" then [CI start] else if [Effect measure] = "Risk Difference" then [#"Events/1000 2.5% CI"] else [CI start]),
    #"Choosing 97.5% CI for MID precision calc" = Table.AddColumn(#"Choosing 2.5% CI for MID precision calc", "97.5% CI for MIDs", each if [Effect measure] = "Risk Ratio" then [CI end] else if [Effect measure] = "Odds Ratio" then [CI end] else if [Effect measure] = "Risk Difference" then [#"Events/1000 97.5% CI"] else [CI end]),
    #"Analyst inputting MIDs" = Table.AddColumn(#"Choosing 97.5% CI for MID precision calc", "MID", each if [Outcome] = "Person/participant generic health-related quality of life - SF-36 physical component score" then 2 else if [Outcome] = "Person/participant generic health-related quality of life - SF-36 mental component score" then 3 else if [Effect measure] = "Risk Ratio" then 0.8 else if [Effect measure] = "Hazard Ratio" then 0.8 else if [Effect measure] = "Odds Ratio" then 0.8 else if [Effect measure] = "Ratio of Means" then 0.8 else null),
    #"Calculating inverse MIDs (assuming linear scale) - for non-linear scale determine differently" = Table.AddColumn(#"Analyst inputting MIDs", "Inverse MID", each [MID] * -1, type number),
    #"Analyst inverse MID choice" = Table.AddColumn(#"Calculating inverse MIDs (assuming linear scale) - for non-linear scale determine differently", "Inverse MID choice", each if [Effect measure] = "Risk Ratio" then 1.25 else if [Effect measure] = "Hazard Ratio" then 1.25 else if [Effect measure] = "Odds Ratio" then 1.25 else if [Effect measure] = "Ratio of Means" then 1.25 else [Inverse MID]),
```

```
    #"2.5% CI relative precision value determination" = Table.AddColumn(#"Analyst inverse MID choice", "2.5% CI MID relation", each if [MID] = null then null else if [#"2.5% CI for MIDs"] < [Inverse MID choice] then -1 else if [#"2.5% CI for MIDs"] < [MID] then 0 else if [#"2.5% CI for MIDs"] >= [MID] then 1 else null),
    #"97.5% CI relative precision value determination" = Table.AddColumn(#"2.5% CI relative precision value determination", "97.5% CI MID relation", each if [MID] = null then null else if [#"97.5% CI for MIDs"] < [Inverse MID choice] then -1 else if [#"97.5% CI for MIDs"] < [MID] then 0 else if [#"97.5% CI for MIDs"] >= [MID] then 1 else null),
    #"Changed Type1" = Table.TransformColumnTypes(#"97.5% CI relative precision value determination",{{"2.5% CI MID relation", Int64.Type}, {"97.5% CI MID relation", Int64.Type}}),
    #"Combining precision CI values together" = Table.AddColumn(#"Changed Type1", "Precision number", each Text.Combine({Text.From([#"2.5% CI MID relation"], "en-GB"), Text.From([#"97.5% CI MID relation"], "en-GB")}, "="), type text),
    #"Replaced Value1" = Table.ReplaceValue(#"Combining precision CI values together","",null,Replacer.ReplaceValue,{"Precision number"}),
    #"Imprecision MID calculation" = Table.AddColumn(#"Replaced Value1", "Imprecision MID", each if [Precision number] = null then null else if [Precision number] = "0=0" then "Not serious" else if [Precision number] = "1=1" then "Not serious" else if [Precision number] = "-1=-1" then "Not serious" else if [Precision number] = "0=1" then "Serious" else if [Precision number] = "1=0" then "Serious" else if [Precision number] = "-1=0" then "Serious" else if [Precision number] = "0=-1" then "Serious" else if [Precision number] = "-1=1" then "Very serious" else if [Precision number] = "1=-1" then "Very serious" else "?"),
    #"Merged Queries" = Table.NestedJoin(#"Imprecision MID calculation", {"Analysis group", "Analysis number"}, #"OIS dichotomous", {"Analysis group", "Analysis number"}, "OIS dichotomous", JoinKind.LeftOuter),
    ///If there were continuous outcomes, you would also merge the query for OIS continuous, get the OIS precision for that, merge that with the OIS precision for dichotomous and then do the rest of the steps as usual
    #"Expanded OIS dichotomous" = Table.ExpandTableColumn(#"Merged Queries", "OIS dichotomous", {"OIS", "OIS Precision"}, {"OIS", "OIS Precision"}),
```

```
    #"Merged Queries1" = Table.NestedJoin(#"Expanded OIS dichotomous",
{"Analysis group", "Analysis number"}, #"OIS continuous", {"Analysis group",
"Analysis number"}, "OIS continuous", JoinKind.LeftOuter),
    #"Expanded OIS continuous" = Table.ExpandTableColumn(#"Merged Queries1",
"OIS continuous", {"OIS", "Precision OIS"}, {"OIS.1", "Precision OIS"}),
    #"Merged Columns" =
Table.CombineColumns(Table.TransformColumnTypes(#"Expanded OIS
continuous", {{"OIS", type text}, {"OIS.1", type text}}, "en-GB"),{"OIS",
"OIS.1"},Combiner.CombineTextByDelimiter("", QuoteStyle.None),"OIS"),
    #"Replaced Value" = Table.ReplaceValue(#"Merged Columns","Not
applicable",null,Replacer.ReplaceValue,{"OIS Precision"}),
    #"Merged Columns1" = Table.CombineColumns(#"Replaced Value",{"Precision
OIS", "OIS Precision"},Combiner.CombineTextByDelimiter("", QuoteStyle.None),"OIS
Precision"),
    #"Replaced Value3" = Table.ReplaceValue(#"Merged
Columns1","",null,Replacer.ReplaceValue,{"OIS Precision", "OIS"}),
    #"Analyst imprecision edit" = Table.AddColumn(#"Replaced Value3", "Imprecision
analyst", each if [Outcome] = "Acute myocardial infarction" then "Very serious" else
null),
    #"Final imprecision calculation" = Table.AddColumn(#"Analyst imprecision edit",
"Imprecision", each if [Imprecision analyst] <> null then [Imprecision analyst] else if
[Imprecision MID] <> null then [Imprecision MID] else if [OIS Precision] <> null then
[OIS Precision] else null),
    #"Method used to determine precision" = Table.AddColumn(#"Final imprecision
calculation", "Method used", each if [Imprecision analyst] <> null then "Imprecision
analyst" else if [Imprecision MID] <> null then "Imprecision MID" else if [OIS
Precision] <> null then "OIS Precision" else null),
    #"Removing not serious" = Table.AddColumn(#"Method used to determine
precision", "Imprecision for reason", each if [Imprecision] = "Not serious" then null
else [Imprecision]),
    #"Removing method for not serious" = Table.AddColumn(#"Removing not
serious", "Method for reason", each if [Imprecision for reason] = null then null else
[Method used]),
```

```
    #"Combining method and imprecision" = Table.AddColumn(#"Removing method for not serious", "Method-Imprecision", each Text.Combine({[Method for reason], [Imprecision for reason]}, "-"), type text),
    #"Replaced Value2" = Table.ReplaceValue(#"Combining method and imprecision","",null,Replacer.ReplaceValue,{"Method-Imprecision"}),
    #"Imprecision reason generation" = Table.AddColumn(#"Replaced Value2", "Imprecision reason", each if [Outcome] = "Mortality" then "Downgraded by 1 interval as agreed with committee" else if [#"Method-Imprecision"] = "Imprecision MID-Serious" then "Serious imprecision because 95%CI crosses 1 MID (RR 0.8-1.25)" else if [#"Method-Imprecision"] = "Imprecision MID-Very serious" then "Very serious imprecision because 95%CI crosses 2 MIDs (RR 0.8-1.25)" else if [#"Method-Imprecision"] = "OIS Precision-Serious" then "Serious imprecision because the number of participants is between 30% and 100% of the optimal information size (OIS)" else if [#"Method-Imprecision"] = "OIS Precision-Very serious" then "Very serious imprecision because the number of participants is less than 30% of the optimal information size (OIS) or the ratio between the confidence intervals exceed 2.5 (odds ratio)/3 (risk ratio) or both" else null),
    #"Removed Other Columns2" = Table.SelectColumns(#"Imprecision reason generation",{"Analysis group", "Analysis number", "Analysis name", "Number of events 1", "n1", "Number of events 2", "n2", "Sample size", "Events/1000", "Events/1000 2.5% CI", "Events/1000 97.5% CI", "SE", "Square Root(N)", "SD", "SD^2", "Events for MIDs", "2.5% CI for MIDs", "97.5% CI for MIDs", "Imprecision", "Imprecision reason"}),
    #"Inserted Merged Column" = Table.AddColumn(#"Removed Other Columns2", "GRADE group", each Text.Combine({Text.From([Analysis group], "en-GB"), Text.From([Analysis number], "en-GB")}, "-"), type text),
    #"Filtered Rows1" = Table.SelectRows(#"Inserted Merged Column", each ([Analysis group] = 1))
in
    #"Filtered Rows1"
```

Number of studies

```
let
```

```
    Source = Excel.Workbook(Web.Contents("source"), null, true),
    #"Data rows_Sheet" = Source{[Item="Data rows",Kind="Sheet"]}[Data],
    #"Promoted Headers" = Table.PromoteHeaders(#"Data rows_Sheet",
[PromoteAllScalars=true]),
    #"Changed Type" = Table.TransformColumnTypes(#"Promoted
Headers",{{"Analysis group", Int64.Type}, {"Analysis number", Int64.Type}, {"Analysis
name", type text}, {"Subgroup", type any}, {"Applicability", type text}, {"Study", type
text}, {"Study year", Int64.Type}, {"GIV Mean", type number}, {"GIV SE", type
number}, {"Experimental mean", type number}, {"Experimental SD", type number},
{"Experimental cases", Int64.Type}, {"Experimental N", Int64.Type}, {"Control mean",
type number}, {"Control SD", type number}, {"Control cases", Int64.Type}, {"Control
N", Int64.Type}, {"O-E", type any}, {"Variance", type any}, {"Weight", type number},
{"Mean", type number}, {"CI start", type number}, {"CI end", type number},
{"Footnotes", type any}}),
    #"Splitting by "" at "" to separate follow up time" = Table.SplitColumn(#"Changed
Type", "Analysis name", Splitter.SplitTextByDelimiter(" at ", QuoteStyle.Csv),
{"Analysis name.1", "Analysis name.2"}),
    #"Changed Type1" = Table.TransformColumnTypes(#"Splitting by "" at "" to
separate follow up time",{{"Analysis name.1", type text}, {"Analysis name.2", type
text}}),
    #"Renamed Columns" = Table.RenameColumns(#"Changed Type1",{{"Analysis
name.2", "Follow up time"}}),
    #"Splitting by "" ("" to separate scale information" = Table.SplitColumn(#"Renamed
Columns", "Analysis name.1", Splitter.SplitTextByDelimiter(" (", QuoteStyle.Csv),
{"Analysis name.1.1", "Analysis name.1.2"}),
    #"Changed Type2" = Table.TransformColumnTypes(#"Splitting by "" ("" to
separate scale information",{{"Analysis name.1.1", type text}, {"Analysis name.1.2",
type text}}),
    #"Renamed Columns1" = Table.RenameColumns(#"Changed Type2",{{"Analysis
name.1.2", "Scale"}, {"Analysis name.1.1", "Protocol Outcome"}}),
    #"Removing closing bracket" = Table.ReplaceValue(#"Renamed
Columns1",")","",Replacer.ReplaceText,{"Scale"}),
    #"Making short outcome" = Table.AddColumn(#"Removing closing bracket",
"Outcome short", each if [Protocol Outcome] = "Mortality" then "Mort" else if [Protocol
```

```
Outcome] = "Physical dependency" then "Pdep" else if [Protocol Outcome] = "Falls" then "Falls" else if [Protocol Outcome] = "Readmissions to hospital" then "Read" else if [Protocol Outcome] = "Person/participant generic health-related quality of life" then "HQOL" else if [Protocol Outcome] = "Activities of daily living" then "ADL" else if [Protocol Outcome] = "Extended activities of daily living" then "EADL" else if [Protocol Outcome] = "Length of hospital stay" then "LOS" else if [Protocol Outcome] = "Caregiver strain index" then "CSI" else if [Protocol Outcome] = "Psychological distress/mood" then "PD" else if [Protocol Outcome] = "Carer generic health-related quality of life" then "CHQOL" else if [Protocol Outcome] = "Stroke-specific Patient-Reported Outcome Measures" then "SQOL" else "?"),
    #"Merged Queries" = Table.NestedJoin(#"Making short outcome", {"Study"}, BiasInd, {"Study"}, "Critical appraisal no edits", JoinKind.LeftOuter),
    #"Expanded Critical appraisal no edits" = Table.ExpandTableColumn(#"Merged Queries", "Critical appraisal no edits", {"Short population", "Short comparison"}, {"Short population", "Short comparison"}),
    #"Removed Duplicates" = Table.Distinct(#"Expanded Critical appraisal no edits"),
    #"Removed Columns" = Table.RemoveColumns(#"Removed Duplicates",{"Footnotes", "Study year", "GIV Mean", "GIV SE", "Experimental mean", "Experimental SD", "Experimental cases", "Experimental N", "Control mean", "Control SD", "Control cases", "Control N", "O-E", "Variance", "Weight", "Mean", "CI start", "CI end", "Outcome short", "Short population", "Short comparison"}),
    #"Duplicated Column" = Table.DuplicateColumn(#"Removed Columns", "Protocol Outcome", "Protocol Outcome - Copy"),
    #"Pivoted Column" = Table.Pivot(#"Duplicated Column", List.Distinct(#"Duplicated Column"[#"Protocol Outcome - Copy"]), "Protocol Outcome - Copy", "Study", List.Count),
    #"Inserted Sum" = Table.AddColumn(#"Pivoted Column", "Number of studies", each List.Sum({[Proportion of patients requiring transfusion], [Thrombotic complications], [Duplicate outcome proportion of people requiring transfusion], [Serious adverse events], [#"All-cause mortality"], [Number of units of allogenic blood transfused], [Infection], [Length of stay], [Acute myocardial infarction], [Volume of allogenic blood transfused], [Total blood loss volume]}), type number),
```

```
    #"Inserted Merged Column" = Table.AddColumn(#"Inserted Sum", "GRADE
group", each Text.Combine({Text.From([Analysis group], "en-GB"),
Text.From([Analysis number], "en-GB")}, "-"), type text),
    #"Removed Columns1" = Table.RemoveColumns(#"Inserted Merged
Column",{"Proportion of patients requiring transfusion", "Thrombotic complications",
"Duplicate outcome proportion of people requiring transfusion", "Serious adverse
events", "All-cause mortality", "Number of units of allogenic blood transfused",
"Infection", "Length of stay", "Acute myocardial infarction", "Volume of allogenic blood
transfused", "Total blood loss volume"})
in
    #"Removed Columns1"
```

Timepoint

```
let
    Source = Excel.Workbook(Web.Contents("source"), null, true),
    #"Data rows_Sheet" = Source{[Item="Data rows",Kind="Sheet"]}[Data],
    #"Promoted Headers" = Table.PromoteHeaders(#"Data rows_Sheet",
[PromoteAllScalars=true]),
    #"Changed Type" = Table.TransformColumnTypes(#"Promoted
Headers",{{"Analysis group", Int64.Type}, {"Analysis number", Int64.Type}, {"Analysis
name", type text}, {"Subgroup", type any}, {"Applicability", type text}, {"Study", type
text}, {"Study year", Int64.Type}, {"GIV Mean", type number}, {"GIV SE", type
number}, {"Experimental mean", type number}, {"Experimental SD", type number},
{"Experimental cases", Int64.Type}, {"Experimental N", Int64.Type}, {"Control mean",
type number}, {"Control SD", type number}, {"Control cases", Int64.Type}, {"Control
N", Int64.Type}, {"O-E", type any}, {"Variance", type any}, {"Weight", type number},
{"Mean", type number}, {"CI start", type number}, {"CI end", type number},
{"Footnotes", type any}}),
    #"Splitting by "" at "" to separate follow up time" = Table.SplitColumn(#"Changed
Type", "Analysis name", Splitter.SplitTextByDelimiter(" at ", QuoteStyle.Csv),
{"Analysis name.1", "Analysis name.2"}),
```

```
    #"Changed Type1" = Table.TransformColumnTypes(#"Splitting by "" at "" to
separate follow up time",{{"Analysis name.1", type text}, {"Analysis name.2", type
text}}),
    #"Renamed Columns" = Table.RenameColumns(#"Changed Type1",{{"Analysis
name.2", "Follow up time"}}),
    #"Splitting by "" ("" to separate scale information" = Table.SplitColumn(#"Renamed
Columns", "Analysis name.1", Splitter.SplitTextByDelimiter(" (", QuoteStyle.Csv),
{"Analysis name.1.1", "Analysis name.1.2"}),
    #"Changed Type2" = Table.TransformColumnTypes(#"Splitting by "" ("" to
separate scale information",{{"Analysis name.1.1", type text}, {"Analysis name.1.2",
type text}}),
    #"Renamed Columns1" = Table.RenameColumns(#"Changed Type2",{{"Analysis
name.1.2", "Scale"}, {"Analysis name.1.1", "Protocol Outcome"}}),
    #"Removing closing bracket" = Table.ReplaceValue(#"Renamed
Columns1",")","",Replacer.ReplaceText,{"Scale"}),
    #"Making short outcome" = Table.AddColumn(#"Removing closing bracket",
"Outcome short", each if [Protocol Outcome] = "Mortality" then "Mort" else if [Protocol
Outcome] = "Physical dependency" then "Pdep" else if [Protocol Outcome] = "Falls"
then "Falls" else if [Protocol Outcome] = "Readmissions to hospital" then "Read" else
if [Protocol Outcome] = "Person/participant generic health-related quality of life" then
"HQOL" else if [Protocol Outcome] = "Activities of daily living" then "ADL" else if
[Protocol Outcome] = "Extended activities of daily living" then "EADL" else if [Protocol
Outcome] = "Length of hospital stay" then "LOS" else if [Protocol Outcome] =
"Caregiver strain index" then "CSI" else if [Protocol Outcome] = "Psychological
distress/mood" then "PD" else if [Protocol Outcome] = "Carer generic health-related
quality of life" then "CHQOL" else if [Protocol Outcome] = "Stroke-specific Patient-
Reported Outcome Measures" then "SQOL" else "?"),
    #"Merged Queries" = Table.NestedJoin(#"Making short outcome", {"Study"},
BiasInd, {"Study"}, "Critical appraisal no edits", JoinKind.LeftOuter),
    #"Expanded Critical appraisal no edits" = Table.ExpandTableColumn(#"Merged
Queries", "Critical appraisal no edits", {"Overall risk of bias", "Overall directness",
"Short population", "Short comparison"}, {"Overall risk of bias", "Overall directness",
"Short population", "Short comparison"}),
    #"Removed Duplicates" = Table.Distinct(#"Expanded Critical appraisal no edits"),
```

#"Removed Columns" = Table.RemoveColumns(#"Removed Duplicates",{"Footnotes", "GIV Mean", "GIV SE", "Experimental mean", "Experimental SD", "Experimental cases", "Experimental N", "Control mean", "Control SD", "Control cases", "Control N", "O-E", "Variance", "Mean", "CI start", "CI end"}),
#"Reordered Columns" = Table.ReorderColumns(#"Removed Columns",{"Analysis group", "Analysis number", "Protocol Outcome", "Scale", "Outcome short", "Short population", "Short comparison", "Follow up time", "Subgroup", "Applicability", "Study", "Study year", "Weight", "Overall risk of bias", "Overall directness"}),
#"Removed Columns1" = Table.RemoveColumns(#"Reordered Columns",{"Follow up time"}),
#"Reordered Columns1" = Table.ReorderColumns(#"Removed Columns1",{"Analysis group", "Analysis number", "Study", "Protocol Outcome", "Scale", "Outcome short", "Short population", "Short comparison", "Subgroup", "Applicability", "Study year", "Weight", "Overall risk of bias", "Overall directness"}),
#"Removed Columns2" = Table.RemoveColumns(#"Reordered Columns1",{"Study year"}),
#"Removed Columns3" = Table.RemoveColumns(#"Removed Columns2",{"Study", "Short population", "Short comparison", "Subgroup", "Applicability"}),
#"Removed Columns4" = Table.RemoveColumns(#"Removed Columns3",{"Weight", "Overall risk of bias", "Overall directness"}),
#"Duplicated Column" = Table.DuplicateColumn(#"Removed Columns4", "Protocol Outcome", "Protocol Outcome - Copy"),
#"Added Custom" = Table.AddColumn(#"Duplicated Column", "Timepoint (months)", each 0),
#"Pivoted Column to calculate average timepoint (months)" = Table.Pivot(#"Added Custom", List.Distinct(#"Added Custom"[#"Protocol Outcome - Copy"]), "Protocol Outcome - Copy", "Timepoint (months)", List.Average),
#"Merged Columns" = Table.CombineColumns(Table.TransformColumnTypes(#"Pivoted Column to calculate average timepoint (months)", {{"Proportion of patients requiring transfusion", type text}, {"Thrombotic complications", type text}, {"Duplicate outcome proportion of people requiring transfusion", type text}, {"Serious adverse events", type text}, {"All-cause mortality", type text}, {"Number of units of allogenic blood

```
transfused", type text}, {"Infection", type text}, {"Length of stay", type text}, {"Acute myocardial infarction", type text}, {"Volume of allogenic blood transfused", type text}, {"Total blood loss volume", type text}}, "en-GB"),{"Proportion of patients requiring transfusion", "Thrombotic complications", "Duplicate outcome proportion of people requiring transfusion", "Serious adverse events", "All-cause mortality", "Number of units of allogenic blood transfused", "Infection", "Length of stay", "Acute myocardial infarction", "Volume of allogenic blood transfused", "Total blood loss volume"},Combiner.CombineTextByDelimiter("", QuoteStyle.None),"Average timepoint (months)"),
    #"Changed Type4" = Table.TransformColumnTypes(#"Merged Columns",{{"Average timepoint (months)", type number}}),
    #"Rounded Off" = Table.TransformColumns(#"Changed Type4",{{"Average timepoint (months)", each Number.Round(_, 1), type number}}),
    #"Inserted Merged Column" = Table.AddColumn(#"Rounded Off", "GRADE code", each Text.Combine({Text.From([Analysis group], "en-GB"), Text.From([Analysis number], "en-GB")}, "-"), type text)
in
    #"Inserted Merged Column"
```

Inconsistency, units

```
let
    Source = Excel.Workbook(Web.Contents("source"), null, true),
    #"Overall estimates_Sheet" = Source{[Item="Overall estimates",Kind="Sheet"]}[Data],
    #"Promoted Headers" = Table.PromoteHeaders(#"Overall estimates_Sheet", [PromoteAllScalars=true]),
    #"Changed Type" = Table.TransformColumnTypes(#"Promoted Headers",{{"Analysis group", Int64.Type}, {"Analysis number", Int64.Type}, {"Analysis name", type text}, {"Analysis group name", type text}, {"Data source", type text}, {"Data source eligibility", type text}, {"Data type", type text}, {"Log-scale data", type logical}, {"Outcome", type text}, {"Intervention grouping", type text}, {"Experimental intervention", type any}, {"Control intervention", type any}, {"Subgroup by", type any}, {"Filter criteria", type any}, {"Experimental group label", type text}, {"Control group
```

```
label", type text}, {"Statistical method", type text}, {"Effect measure", type text}, {"Unit of effect measure", type any}, {"Analysis model", type text}, {"Heterogeneity estimator", type any}, {"Tau² CI", type logical}, {"Subgroup estimates", type logical}, {"Overall estimates", type logical}, {"Test for subgroup differences", type logical}, {"Prediction interval", type logical}, {"Swap event and non-event", type logical}, {"CI method", type text}, {"CI/PI level", type number}, {"Experimental cases", Int64.Type}, {"Experimental N", Int64.Type}, {"Control cases", Int64.Type}, {"Control N", Int64.Type}, {"Mean", type number}, {"CI start", type number}, {"CI end", type number}, {"PI start", type any}, {"PI end", type any}, {"Heterogeneity Tau²", type any}, {"Tau² CI start", type any}, {"Tau² CI end", type any}, {"Heterogeneity Chi²", type number}, {"Heterogeneity df", Int64.Type}, {"Heterogeneity P", type number}, {"Heterogeneity I²", type number}, {"Effect Z", type number}, {"Effect P", type number}, {"Subgroup Chi²", type any}, {"Subgroup df", type any}, {"Subgroup P", type any}, {"Subgroup I²", type any}, {"ID", type text}}),
    #"Removed Columns" = Table.RemoveColumns(#"Changed Type",{"PI start", "PI end", "Heterogeneity Tau²", "Tau² CI start", "Tau² CI end", "Heterogeneity Chi²", "Heterogeneity df", "Heterogeneity P", "Effect Z", "Effect P", "Subgroup Chi²", "Subgroup df", "Subgroup P", "ID", "CI method", "Test for subgroup differences", "Prediction interval", "Swap event and non-event", "Tau² CI", "Heterogeneity estimator", "Experimental intervention", "Control intervention"}),
    #"Inconsistency calculation" = Table.AddColumn(#"Removed Columns", "Inconsistency calculation", each if [#"Heterogeneity I²"] = null then "Serious - single study?" else if [#"Heterogeneity I²"] < 0 then "?" else if [#"Heterogeneity I²"] < 40 then "Not serious" else if [#"Heterogeneity I²"] < 60 then "Serious" else if [#"Heterogeneity I²"] < 80 then "Very serious" else if [#"Heterogeneity I²"] < 100 then "Very serious - too heterogenous to meaningfully pool?" else "?"),
    #"Analyst edit of inconsistency" = Table.AddColumn(#"Inconsistency calculation", "Inconsistency analyst check", each if [Analysis name] = "Volume of allogenic blood transfused (mL) at end of study" then "Serious" else if [Analysis name] = "Acute myocardial infarction at end of study" then "Serious" else if [Analysis name] = "Length of stay (days) at end of study" then "Very serious" else if [Analysis name] = "Total blood loss volume at end of study" then "Very serious" else null),
```

```
    #"Final inconsistency calculation" = Table.AddColumn(#"Analyst edit of
inconsistency", "Inconsistency", each if [Inconsistency analyst check] <> null then
[Inconsistency analyst check] else [Inconsistency calculation]),
    #"Inconsistency reason" = Table.AddColumn(#"Final inconsistency calculation",
"Inconsistency reason", each if Text.Contains([Inconsistency calculation], "single
study") then "Inconsistency: Single study- downgraded once for inconsistency, as
single study outcomes may otherwise receive favourable ratings for inconsistency by
default" else if [#"Heterogeneity I²"] < 40 then null else if [#"Heterogeneity I²"] < 60
then "Inconsistency: Downgraded once. Serious heterogeneity (I2 = 40 to 60%)
unexplained by subgroup analysis. Random effects analysis used" else if
[#"Heterogeneity I²"] < 80 then "Inconsistency: Downgraded twice. Very serious
heterogeneity (serious I2 = >60%) unexplained by subgroup analysis. Random
effects analysis used" else if [#"Heterogeneity I²"] <= 100 then "Inconsistency:
Downgraded twice. Very serious heterogeneity (serious I2 = >60%) unexplained by
subgroup analysis. Random effects analysis used" else if [#"Heterogeneity I²"] > 100
then "?" else if [Analysis name] = "Mortality" then null else null),
    #"Removed Columns1" = Table.RemoveColumns(#"Inconsistency
reason",{"Inconsistency calculation", "Inconsistency analyst check"}),
    #"Inserted Merged Column" = Table.AddColumn(#"Removed Columns1", "GRADE
group", each Text.Combine({Text.From([Analysis group], "en-GB"),
Text.From([Analysis number], "en-GB")}, "-"), type text),
    #"Analyst editable units column" = Table.AddColumn(#"Inserted Merged Column",
"Units", each if [Unit of effect measure] <> null then [Unit of effect measure] else if
[Outcome] = "Length of stay" then "days" else if [Outcome] = "Number of units of
allogenic blood transfused" then "units" else if [Outcome] = "Volume of allogenic
blood transfused" then "mL" else "Not applicable"),
    #"Added Custom1" = Table.AddColumn(#"Analyst editable units column", "|", each
"|"),
    #"Analyst editable study type" = Table.AddColumn(#"Added Custom1", "Study
type", each if [Outcome] = "a" then "Non-randomised studies" else if [Outcome] = "b"
then "Randomised trials" else "Randomised trials"),
    #"Analyst editable ROBINS-I used?" = Table.AddColumn(#"Analyst editable study
type", "ROBINS-I used?", each if [Study type] = "Randomised trials" then false else if
[Outcome] = "a" then true else if [Outcome] = "b" then false else true),
```

```
    #"Analyst editable publication bias" = Table.AddColumn(#"Analyst editable
ROBINS-I used?", "Publication bias", each if [Outcome] = "a" then "Undetected" else
if [Outcome] = "b" then "Strongly suspected" else "Undetected"),
    #"Analyst editable large effect" = Table.AddColumn(#"Analyst editable publication
bias", "Large effect", each if [Study type] = "Randomised trials" then "No" else if
[Outcome] = "a" then "No" else if [Outcome] = "b" then "Large" else if [Outcome] = "c"
then "Very large" else "No"),
    #"Analyst editable plausible confounding" = Table.AddColumn(#"Analyst editable
large effect", "Plausible confounding", each if [Study type] = "Randomised trials" then
"No" else if [Outcome] = "a" then "No" else if [Outcome] = "b" then "Would reduce
demonstrated effect" else if [Outcome] = "c" then "Would suggest spurious effect"
else "No"),
    #"Analyst editable dose response gradient" = Table.AddColumn(#"Analyst editable
plausible confounding", "Dose response gradient", each if [Study type] =
"Randomised trials" then "No" else if [Outcome] = "a" then "No" else if [Outcome] =
"b" then "No" else "No"),
    #"Making other considerations column" = Table.AddColumn(#"Analyst editable
dose response gradient", "Other considerations", each Text.Combine({[Publication
bias], [Large effect], [Plausible confounding], [Dose response gradient]}, "
"), type text),
    #"Replaced Value" = Table.ReplaceValue(#"Making other considerations
column","Undetected
No
No
No","None",Replacer.ReplaceText,{"Other considerations"}),
    #"Analyst editable importance" = Table.AddColumn(#"Replaced Value",
"Importance", each if [Outcome] = "Health-related quality of life" then "Critical" else if
[Outcome] = "Mortality" then "Critical" else if [Outcome] = "a" then "Critical" else if
[Outcome] = "b" then "Important" else "Critical"),
    #"Removed Columns2" = Table.RemoveColumns(#"Analyst editable
importance",{"Publication bias", "Large effect", "Plausible confounding", "Dose
response gradient"})
in
    #"Removed Columns2"
```

Median baseline rate

```
let
    Source = Excel.Workbook(Web.Contents("source"), null, true),
    #"Data rows_Sheet" = Source{[Item="Data rows",Kind="Sheet"]}[Data],
    #"Promoted Headers" = Table.PromoteHeaders(#"Data rows_Sheet",
[PromoteAllScalars=true]),
    #"Changed Type" = Table.TransformColumnTypes(#"Promoted
Headers",{{"Analysis group", Int64.Type}, {"Analysis number", Int64.Type}, {"Analysis
name", type text}, {"Subgroup", type any}, {"Applicability", type text}, {"Study", type
text}, {"Study year", Int64.Type}, {"GIV Mean", type number}, {"GIV SE", type
number}, {"Experimental mean", type number}, {"Experimental SD", type number},
{"Experimental cases", Int64.Type}, {"Experimental N", Int64.Type}, {"Control mean",
type number}, {"Control SD", type number}, {"Control cases", Int64.Type}, {"Control
N", Int64.Type}, {"O-E", type any}, {"Variance", type any}, {"Weight", type number},
{"Mean", type number}, {"CI start", type number}, {"CI end", type number},
{"Footnotes", type any}}),
    #"Splitting by "" at "" to separate follow up time" = Table.SplitColumn(#"Changed
Type", "Analysis name", Splitter.SplitTextByDelimiter(" at ", QuoteStyle.Csv),
{"Analysis name.1", "Analysis name.2"}),
    #"Changed Type1" = Table.TransformColumnTypes(#"Splitting by "" at "" to
separate follow up time",{{"Analysis name.1", type text}, {"Analysis name.2", type
text}}),
    #"Renamed Columns" = Table.RenameColumns(#"Changed Type1",{{"Analysis
name.2", "Follow up time"}}),
    #"Splitting by "" ("" to separate scale information" = Table.SplitColumn(#"Renamed
Columns", "Analysis name.1", Splitter.SplitTextByDelimiter(" (", QuoteStyle.Csv),
{"Analysis name.1.1", "Analysis name.1.2"}),
    #"Changed Type2" = Table.TransformColumnTypes(#"Splitting by "" ("" to
separate scale information",{{"Analysis name.1.1", type text}, {"Analysis name.1.2",
type text}}),
    #"Renamed Columns1" = Table.RenameColumns(#"Changed Type2",{{"Analysis
name.1.2", "Scale"}, {"Analysis name.1.1", "Protocol Outcome"}}),
```

```
    #"Removing closing bracket" = Table.ReplaceValue(#"Renamed
Columns1",")","",Replacer.ReplaceText,{"Scale"}),
    #"Removed Duplicates" = Table.Distinct(#"Removing closing bracket"),
    #"Inserted Division" = Table.AddColumn(#"Removed Duplicates", "Control group
rate", each [Control cases] / [Control N], type number),
    #"Removed Columns" = Table.RemoveColumns(#"Inserted Division",{"Footnotes",
"O-E", "Variance", "Weight", "Mean", "CI start", "CI end", "GIV Mean", "GIV SE",
"Experimental mean", "Experimental SD", "Experimental cases", "Experimental N",
"Control mean", "Control SD", "Study year", "Study", "Control cases", "Control N"}),
    #"Duplicated Column" = Table.DuplicateColumn(#"Removed Columns", "Protocol
Outcome", "Protocol Outcome - Copy"),
    #"Pivoted Column" = Table.Pivot(#"Duplicated Column", List.Distinct(#"Duplicated
Column"[#"Protocol Outcome - Copy"]), "Protocol Outcome - Copy", "Control group
rate", List.Average),
    #"Combining median control events" =
Table.CombineColumns(Table.TransformColumnTypes(#"Pivoted Column",
{{"Proportion of patients requiring transfusion", type text}, {"All-cause mortality", type
text}, {"Length of stay", type text}, {"Number of units of allogenic blood transfused",
type text}, {"Volume of allogenic blood transfused", type text}, {"Serious adverse
events", type text}, {"Thrombotic complications", type text}, {"Acute myocardial
infarction", type text}, {"Infection", type text}, {"Duplicate outcome proportion of
people requiring transfusion", type text}, {"Total blood loss volume", type text}}, "en-
GB"),{"Proportion of patients requiring transfusion", "All-cause mortality", "Length of
stay", "Number of units of allogenic blood transfused", "Volume of allogenic blood
transfused", "Serious adverse events", "Thrombotic complications", "Acute
myocardial infarction", "Infection", "Duplicate outcome proportion of people requiring
transfusion", "Total blood loss volume"},Combiner.CombineTextByDelimiter("",
QuoteStyle.None),"Median control group rate"),
    #"Changed Type3" = Table.TransformColumnTypes(#"Combining median control
events",{{"Median control group rate", type number}}),
    #"Producing GRADE number" = Table.AddColumn(#"Changed Type3", "GRADE
number", each Text.Combine({Text.From([Analysis group], "en-GB"),
Text.From([Analysis number], "en-GB")}, "-"), type text)
in
```

```
    #"Producing GRADE number"
```

OIS continuous

```
let
    Source = Excel.Workbook(Web.Contents("source"), null, true),
    #"Overall estimates_Sheet" = Source{[Item="Data rows",Kind="Sheet"]}[Data],
    #"Promoted Headers" = Table.PromoteHeaders(#"Overall estimates_Sheet",
[PromoteAllScalars=true]),
    #"Changed Type" = Table.TransformColumnTypes(#"Promoted
Headers",{{"Analysis group", Int64.Type}, {"Analysis number", Int64.Type}, {"Analysis
name", type text}, {"Study year", Int64.Type}, {"GIV Mean", type number}, {"GIV SE",
type number}, {"Experimental mean", type number}, {"Experimental SD", type
number}, {"Experimental cases", Int64.Type}, {"Experimental N", Int64.Type},
{"Control mean", type number}, {"Control SD", type number}, {"Control cases",
Int64.Type}, {"Control N", Int64.Type}, {"O-E", type any}, {"Variance", type any},
{"Weight", type number}, {"Mean", type number}, {"CI start", type number}, {"CI end",
type number}, {"Footnotes", type any}, {"Study", type text}, {"Applicability", type text},
{"Subgroup number", type text}, {"Subgroup", type text}}),
    #"Removed Other Columns" = Table.SelectColumns(#"Changed Type",{"Analysis
group", "Analysis number", "Analysis name", "Subgroup", "Subgroup number",
"Applicability", "Study", "Study year", "GIV Mean", "GIV SE", "Experimental mean",
"Experimental SD", "Experimental cases", "Experimental N", "Control mean", "Control
SD", "Control cases", "Control N", "O-E", "Variance", "Weight", "Mean", "CI start", "CI
end"}),
    #"Filtered Rows" = Table.SelectRows(#"Removed Other Columns", each
([Experimental mean] <> null) and ([Analysis name] <> "Total blood loss volume at
end of study")),
    #"1" = Table.AddColumn(#"Filtered Rows", "1", each 1),
    #"Replace weight value ""0"" with ""100""" =
Table.ReplaceValue(#"1",0,100,Replacer.ReplaceValue,{"Weight"}),
    #"Reciprocal experimental n" = Table.AddColumn(#"Replace weight value ""0""
with ""100""", "1/n1", each [1] / [Experimental N], type number),
```

```
    #"Reciprocal control n" = Table.AddColumn(#"Reciprocal experimental n", "1/n2",
each [1] / [Control N], type number),
    #"Adding reciprocal n" = Table.AddColumn(#"Reciprocal control n", "1/n1+1/n2",
each [#"1/n1"] + [#"1/n2"], type number),
    #"Calculated Square Root" = Table.TransformColumns(#"Adding reciprocal
n",{{"1/n1+1/n2", Number.Sqrt, type number}}),
    #"Calculate GIV SD by dividing SE by SQRT reciprocal n" =
Table.AddColumn(#"Calculated Square Root", "SD (GIV)", each [GIV SE] /
[#"1/n1+1/n2"], type number),
    #"Add the SDs" = Table.AddColumn(#"Calculate GIV SD by dividing SE by SQRT
reciprocal n", "SD (general)", each [Experimental SD] + [Control SD], type number),
    #"Divide to work out the average SD" = Table.TransformColumns(#"Add the SDs",
{{"SD (general)", each _ / 2, type number}}),
    #"Condition decide which SD to use" = Table.AddColumn(#"Divide to work out the
average SD", "Common SD", each if [Experimental SD] = 0 then [#"SD (GIV)"] else
[#"SD (general)"]),
    #"Changed Type1" = Table.TransformColumnTypes(#"Condition decide which SD
to use",{{"Common SD", type number}}),
    #"Calculate mean difference" = Table.AddColumn(#"Changed Type1", "Mean
difference calc", each [Experimental mean] - [Control mean], type number),
    #"Condition decide which MD to use" = Table.AddColumn(#"Calculate mean
difference", "MD", each if [Experimental SD] = 0 then [GIV Mean] else [Mean
difference calc]),
    #"Calculate SMD" = Table.AddColumn(#"Condition decide which MD to use",
"SMD", each [MD] / [Common SD], type number),
    #"sqrt(2)" = Table.AddColumn(#"Calculate SMD", "sqrt(2)", each Number.Sqrt(2),
type number),
    #"SD * sqrt(2)" = Table.AddColumn(#"sqrt(2)", "Bottom", each [Common SD] *
[#"sqrt(2)"], type number),
    #"Calculate SMD prep" = Table.AddColumn(#"SD * sqrt(2)", "SMDprep", each
[MD] / [Bottom], type number),
    #"Divided Column" = Table.TransformColumns(#"Calculate SMD prep", {{"Weight",
each _ / 100, type number}}),
```

```
    #"Calculate weighted SMD prep" = Table.AddColumn(#"Divided Column",
"Weighted SMDprep", each [Weight] * [SMDprep], type number),
    #"Removed Other Columns1" = Table.SelectColumns(#"Calculate weighted SMD
prep",{"Analysis group", "Analysis number", "Analysis name", "Subgroup", "Subgroup
number", "Experimental N", "Control N", "Weighted SMDprep"}),
    #"Unpivoted Columns" = Table.UnpivotOtherColumns(#"Removed Other
Columns1", {"Analysis group", "Analysis number", "Analysis name", "Subgroup",
"Subgroup number"}, "Attribute", "Value"),
    #"Pivoting using sum to get SMD for outcome" = Table.Pivot(#"Unpivoted
Columns", List.Distinct(#"Unpivoted Columns"[Attribute]), "Attribute", "Value",
List.Sum),
    #"Absolute value only for SMD" = Table.TransformColumns(#"Pivoting using sum
to get SMD for outcome",{{"Weighted SMDprep", Number.Abs, type number}}),
    #"Rename to nonctr" = Table.RenameColumns(#"Absolute value only for
SMD",{{"Weighted SMDprep", "nonctr"}}),
    #"za (assuming power of 0.8)" = Table.AddColumn(#"Rename to nonctr", "za",
each 0.841621234),
    #"zb (assuming 2 tailed test, alpha of 0.05)" = Table.AddColumn(#"za (assuming
power of 0.8)", "zb", each 1.959963985),
    #"za+zb" = Table.AddColumn(#"zb (assuming 2 tailed test, alpha of 0.05)",
"za+zb", each [za] + [zb], type number),
    #"Divide z complex by nonctr" = Table.AddColumn(#"za+zb", "sss", each
[#"za+zb"] / [nonctr], type number),
    Squared = Table.AddColumn(#"Divide z complex by nonctr", "OIS", each
Number.Power([sss], 2), type number),
    #"Rounded to make OIS" = Table.TransformColumns(Squared,{{"OIS", each
Number.Round(_, 0), type number}}),
    #"Multiply by 2 to get OIS for both arms" = Table.TransformColumns(#"Rounded to
make OIS", {{"OIS", each _ * 2, type number}}),
    #"Calculate total N" = Table.AddColumn(#"Multiply by 2 to get OIS for both arms",
"Total N", each [Experimental N] + [Control N], type number),
    #"Inserted Division" = Table.AddColumn(#"Calculate total N", "Prop OIS", each
[Total N] / [OIS], type number),
```

```
    #"Added Conditional Column" = Table.AddColumn(#"Inserted Division",
"Precision", each if [Prop OIS] >= 1 then "No concerns regarding optimal information
size" else if [Total N] >= 800 then "No concerns regarding optimal information size"
else if [Prop OIS] >= 0.3 then "Serious concerns regarding optimal information size"
else if [Prop OIS] < 0.3 then "Very serious concerns regarding optimal information
size" else "?"),
    #"Added Conditional Column1" = Table.AddColumn(#"Added Conditional Column",
"Precision OIS", each if [Prop OIS] >= 1 then "Not serious" else if [Total N] >= 800
then "Not serious" else if [Prop OIS] >= 0.3 then "Serious" else if [Prop OIS] < 0.3
then "Very serious" else "Not serious"),
    #"Removed Other Columns2" = Table.SelectColumns(#"Added Conditional
Column1",{"Analysis group", "Analysis number", "Analysis name", "Subgroup",
"Subgroup number", "Experimental N", "Control N", "OIS", "Total N", "Precision",
"Precision OIS"})
in
    #"Removed Other Columns2"
```

OIS dichotomous

```
let
    Source = Excel.Workbook(Web.Contents("source"), null, true),
    #"Overall estimates_Sheet" = Source{[Item="Overall
estimates",Kind="Sheet"]}[Data],
    #"Promoted Headers" = Table.PromoteHeaders(#"Overall estimates_Sheet",
[PromoteAllScalars=true]),
    #"Changed Type" = Table.TransformColumnTypes(#"Promoted
Headers",{{"Analysis group", Int64.Type}, {"Analysis number", Int64.Type}, {"Analysis
name", type text}, {"Analysis group name", type text}, {"Data source", type text},
{"Data source eligibility", type text}, {"Data type", type text}, {"Log-scale data", type
logical}, {"Outcome", type text}, {"Intervention grouping", type text}, {"Experimental
intervention", type any}, {"Control intervention", type any}, {"Subgroup by", type any},
{"Filter criteria", type any}, {"Experimental group label", type text}, {"Control group
label", type text}, {"Statistical method", type text}, {"Effect measure", type text}, {"Unit
of effect measure", type any}, {"Analysis model", type text}, {"Heterogeneity
```

estimator", type any}, {"Tau² CI", type logical}, {"Subgroup estimates", type logical}, {"Overall estimates", type logical}, {"Test for subgroup differences", type logical}, {"Prediction interval", type logical}, {"Swap event and non-event", type logical}, {"CI method", type text}, {"CI/PI level", type number}, {"Experimental cases", Int64.Type}, {"Experimental N", Int64.Type}, {"Control cases", Int64.Type}, {"Control N", Int64.Type}, {"Mean", type number}, {"CI start", type number}, {"CI end", type number}, {"PI start", type any}, {"PI end", type any}, {"Heterogeneity Tau²", type any}, {"Tau² CI start", type any}, {"Tau² CI end", type any}, {"Heterogeneity Chi²", type number}, {"Heterogeneity df", Int64.Type}, {"Heterogeneity P", type number}, {"Heterogeneity I²", type number}, {"Effect Z", type number}, {"Effect P", type number}, {"Subgroup Chi²", type any}, {"Subgroup df", type any}, {"Subgroup P", type any}, {"Subgroup I²", type any}, {"ID", type text}}),

#"Removed Other Columns" = Table.SelectColumns(#"Changed Type",{"Analysis group", "Analysis number", "Analysis name", "Analysis group name", "Outcome", "Intervention grouping", "Experimental intervention", "Control intervention", "Experimental group label", "Control group label", "Effect measure", "Experimental cases", "Experimental N", "Control cases", "Control N", "Mean", "CI start", "CI end", "ID"}),

#"Removing mean difference and std. mean difference" = Table.SelectRows(#"Removed Other Columns", each [Effect measure] <> "Mean Difference" and [Effect measure] <> "Std. Mean Difference"),

#"Reordered Columns" = Table.ReorderColumns(#"Removing mean difference and std. mean difference",{"Analysis group", "Analysis number", "ID", "Analysis name", "Analysis group name", "Outcome", "Intervention grouping", "Experimental intervention", "Control intervention", "Experimental group label", "Control group label", "Effect measure", "Experimental cases", "Experimental N", "Control cases", "Control N"}),

#"Calculate proportion experimental group" = Table.AddColumn(#"Reordered Columns", "p1", each [Experimental cases] / [Experimental N], type number),

#"Calculate proportion control group" = Table.AddColumn(#"Calculate proportion experimental group", "p2", each [Control cases] / [Control N], type number),

#"Proportion unexposed with outcomes for GIV" = Table.AddColumn(#"Calculate proportion control group", "prop unexposed", each 0.5),

```
    #"Risk ratios only GIV" = Table.AddColumn(#"Proportion unexposed with outcomes for GIV", "Risk ratio", each if [Effect measure] = "Risk Ratio" then [Mean] else null),
    #"Odds ratios only GIV" = Table.AddColumn(#"Risk ratios only GIV", "Odds ratio", each if [Effect measure] = "Odds Ratio" then [Mean] else null),
    #"Risk differences only GIV" = Table.AddColumn(#"Odds ratios only GIV", "Risk difference", each if [Effect measure] = "Risk difference" then [Mean] else null),
    #"Changed Type2" = Table.TransformColumnTypes(#"Risk differences only GIV",{{"prop unexposed", type number}, {"Risk ratio", type number}, {"Odds ratio", type number}, {"Risk difference", type number}}),
    #"OR-1" = Table.AddColumn(#"Changed Type2", "OR-1", each [Odds ratio] - 1, type number),
    #"proportion unexposed+1" = Table.AddColumn(#"OR-1", "1+pe", each [prop unexposed] + 1, type number),
    #"Top equation portion" = Table.AddColumn(#"proportion unexposed+1", "OR*prop unexposed", each [prop unexposed] * [Odds ratio], type number),
    #"Bottom equation portion" = Table.AddColumn(#"Top equation portion", "(OR-1)*(1+pe)", each [#"1+pe"] * [#"OR-1"], type number),
    #"Odds ratio proportion exposed GIV" = Table.AddColumn(#"Bottom equation portion", "OR prop exposed", each [#"OR*prop unexposed"] / [#"(OR-1)*(1+pe)"], type number),
    #"Risk ratio proportion exposed GIV" = Table.AddColumn(#"Odds ratio proportion exposed GIV", "RR prop exposed", each [prop unexposed] * [Risk ratio], type number),
    #"Risk difference proportion exposed GIV" = Table.AddColumn(#"Risk ratio proportion exposed GIV", "RD prop exposed", each [Risk difference] + [prop unexposed], type number),
    #"Added Conditional Column" = Table.AddColumn(#"Risk difference proportion exposed GIV", "Proportion exposed", each if [Effect measure] = "Risk Ratio" then [RR prop exposed] else if [Effect measure] = "Odds Ratio" then [OR prop exposed] else if [Effect measure] = "Risk Difference" then [RD prop exposed] else null),
    #"Renamed Columns" = Table.RenameColumns(#"Added Conditional Column",{{"p1", "p1 pre"}, {"p2", "p2 pre"}}),
```

```
    #"Added Conditional Column1" = Table.AddColumn(#"Renamed Columns", "p1",
each if [p1 pre] = null then [Proportion exposed] else [p1 pre]),
    #"Added Conditional Column2" = Table.AddColumn(#"Added Conditional
Column1", "p2", each if [p2 pre] = null then [prop unexposed] else [p2 pre]),
    #"Adding proportions" = Table.AddColumn(#"Added Conditional Column2", "pb",
each [p1] + [p2], type number),
    #"Divide added proportions by 2" = Table.TransformColumns(#"Adding
proportions", {{"pb", each _ / 2, type number}}),
    #"1" = Table.AddColumn(#"Divide added proportions by 2", "1", each 1),
    #"1-pb" = Table.AddColumn(#"1", "qb", each [1] - [pb], type number),
    #"1-p1" = Table.AddColumn(#"1-pb", "q1", each [1] - [p1], type number),
    #"1-p2" = Table.AddColumn(#"1-p1", "q2", each [1] - [p2], type number),
    #"p1-p2" = Table.AddColumn(#"1-p2", "Delta", each [p1] - [p2], type number),
    #"Absolute only (delta)" = Table.TransformColumns(#"p1-p2",{{"Delta",
Number.Abs, type number}}),
    #"za (assuming power 0.8)" = Table.AddColumn(#"Absolute only (delta)", "za",
each 0.841621234),
    #"zb (assuming 2 tailed test, alpha 0.05)" = Table.AddColumn(#"za (assuming
power 0.8)", "zb", each 1.959963985),
    #"2" = Table.AddColumn(#"zb (assuming 2 tailed test, alpha 0.05)", "2", each 2),
    #"2pq" = Table.AddColumn(#"2", "sqrt2pq", each List.Product({[pb], [qb], [2]}), type
number),
    #"sqrt(2pq)" = Table.TransformColumns(#"2pq",{{"sqrt2pq", Number.Sqrt, type
number}}),
    p1q1 = Table.AddColumn(#"sqrt(2pq)", "p1q1", each [p1] * [q1], type number),
    p2q2 = Table.AddColumn(p1q1, "p2q2", each [p2] * [q2], type number),
    sumpqs = Table.AddColumn(p2q2, "sqrtsumpqs", each [p1q1] + [p2q2], type
number),
    #"sqrt(sumpqs)" = Table.TransformColumns(sumpqs,{{"sqrtsumpqs",
Number.Sqrt, type number}}),
    #"sqrt2pq * za" = Table.AddColumn(#"sqrt(sumpqs)", "sqrt2pq*za", each [sqrt2pq]
* [za], type number),
    #"sqrtsumpqs * zb" = Table.AddColumn(#"sqrt2pq * za", "sqrtsumpqs*zb", each
[sqrtsumpqs] * [zb], type number),
```

#"Adding the two to make the top" = Table.AddColumn(#"sqrtsumpqs * zb", "Top", each [#"sqrt2pq*za"] + [#"sqrtsumpqs*zb"], type number),

#"Dividing the top by delta" = Table.AddColumn(#"Adding the two to make the top", "sss", each [Top] / [Delta], type number),

#"Squaring the value" = Table.AddColumn(#"Dividing the top by delta", "OIS", each Number.Power([sss], 2), type number),

#"Rounding it up to make the OIS" = Table.TransformColumns(#"Squaring the value",{{"OIS", each Number.Round(_, 0), type number}}),

#"Removed Other Columns1" = Table.SelectColumns(#"Rounding it up to make the OIS",{"Analysis group", "Analysis number", "Analysis name", "Analysis group name", "Outcome", "Intervention grouping", "Experimental intervention", "Control intervention", "Experimental group label", "Control group label", "Effect measure", "Experimental cases", "Experimental N", "Control cases", "CI start", "CI end", "Control N", "OIS"}),

#"Inserted Addition" = Table.AddColumn(#"Removed Other Columns1", "Total N", each [Experimental N] + [Control N], Int64.Type),

#"Inserted Multiplication" = Table.AddColumn(#"Inserted Addition", "OIS two arms", each [OIS] * 2, type number),

#"OIS proportion calculation" = Table.AddColumn(#"Inserted Multiplication", "Prop OIS", each [Total N] / [OIS two arms], type number),

#"CI proportion calculation" = Table.AddColumn(#"OIS proportion calculation", "Prop CI", each [CI end] / [CI start], type number),

#"RR CI proportion conditional" = Table.AddColumn(#"CI proportion calculation", "RR Prop CI", each if [Effect measure] = "Risk Ratio" then [Prop CI] else null),

#"OR CI proportion conditional" = Table.AddColumn(#"RR CI proportion conditional", "OR Prop CI", each if [Effect measure] = "Odds Ratio" then [Prop CI] else null),

#"For RR and OR proportions - replace null with 0" = Table.ReplaceValue(#"OR CI proportion conditional",null,0,Replacer.ReplaceValue,{"RR Prop CI", "OR Prop CI"}),

#"Changed Type1" = Table.TransformColumnTypes(#"For RR and OR proportions - replace null with 0",{{"RR Prop CI", type number}, {"OR Prop CI", type number}}),

#"Precision conditional determination" = Table.AddColumn(#"Changed Type1", "Precision", each if [Effect measure] = "Hazard Ratio" then "Cannot be determined using this method" else if [Effect measure] = "Ratio of Means" then "Cannot be

```
determined using this method" else if [RR Prop CI] >= 3 then "Very serious concerns regarding optimal information size" else if [OR Prop CI] >= 2.5 then "Very serious concerns regarding optimal information size" else if [Prop OIS] >= 1 then "No concerns regarding optimal information size" else if [Prop OIS] >= 0.3 then "Serious concerns regarding optimal information size" else if [Prop OIS] < 0.3 then "Very serious concerns regarding optimal information size" else "?"),
    Custom1 = Table.AddColumn(#"Precision conditional determination", "OIS Precision", each if [Effect measure] = "Hazard Ratio" then "Not applicable" else if [Effect measure] = "Ratio of Means" then "Not applicable" else if [RR Prop CI] >= 3 then "Very serious" else if [OR Prop CI] >= 2.5 then "Very serious" else if [Prop OIS] >= 1 then "Not serious" else if [Prop OIS] >= 0.3 then "Serious" else if [Prop OIS] < 0.3 then "Very serious" else "Not serious")
in
    Custom1
```

GRADE table

```
let
    Source = Excel.Workbook(Web.Contents("source"), null, true),
    #"Data rows_Sheet" = Source{[Item="Data rows",Kind="Sheet"]}[Data],
    #"Promoted Headers" = Table.PromoteHeaders(#"Data rows_Sheet", [PromoteAllScalars=true]),
    #"Changed Type" = Table.TransformColumnTypes(#"Promoted Headers",{{"Analysis group", Int64.Type}, {"Analysis number", Int64.Type}, {"Analysis name", type text}, {"Subgroup", type any}, {"Applicability", type text}, {"Study", type text}, {"Study year", Int64.Type}, {"GIV Mean", type number}, {"GIV SE", type number}, {"Experimental mean", type number}, {"Experimental SD", type number}, {"Experimental cases", Int64.Type}, {"Experimental N", Int64.Type}, {"Control mean", type number}, {"Control SD", type number}, {"Control cases", Int64.Type}, {"Control N", Int64.Type}, {"O-E", type any}, {"Variance", type any}, {"Weight", type number}, {"Mean", type number}, {"CI start", type number}, {"CI end", type number}, {"Footnotes", type any}}),
```

```
    #"Splitting by "" at "" to separate follow up time" = Table.SplitColumn(#"Changed
Type", "Analysis name", Splitter.SplitTextByDelimiter(" at ", QuoteStyle.Csv),
{"Analysis name.1", "Analysis name.2"}),
    #"Changed Type1" = Table.TransformColumnTypes(#"Splitting by "" at "" to
separate follow up time",{{"Analysis name.1", type text}, {"Analysis name.2", type
text}}),
    #"Renamed Columns" = Table.RenameColumns(#"Changed Type1",{{"Analysis
name.2", "Follow up time"}}),
    #"Splitting by "" ("" to separate scale information" = Table.SplitColumn(#"Renamed
Columns", "Analysis name.1", Splitter.SplitTextByDelimiter(" (", QuoteStyle.Csv),
{"Analysis name.1.1", "Analysis name.1.2"}),
    #"Changed Type2" = Table.TransformColumnTypes(#"Splitting by "" ("" to
separate scale information",{{"Analysis name.1.1", type text}, {"Analysis name.1.2",
type text}}),
    #"Renamed Columns1" = Table.RenameColumns(#"Changed Type2",{{"Analysis
name.1.2", "Scale"}, {"Analysis name.1.1", "Protocol Outcome"}}),
    #"Removing closing bracket" = Table.ReplaceValue(#"Renamed
Columns1",")","",Replacer.ReplaceText,{"Scale"}),
    #"Making short outcome" = Table.AddColumn(#"Removing closing bracket",
"Outcome short", each if [Protocol Outcome] = "Mortality" then "Mort" else if [Protocol
Outcome] = "Physical dependency" then "Pdep" else if [Protocol Outcome] = "Falls"
then "Falls" else if [Protocol Outcome] = "Readmissions to hospital" then "Read" else
if [Protocol Outcome] = "Person/participant generic health-related quality of life" then
"HQOL" else if [Protocol Outcome] = "Activities of daily living" then "ADL" else if
[Protocol Outcome] = "Extended activities of daily living" then "EADL" else if [Protocol
Outcome] = "Length of hospital stay" then "LOS" else if [Protocol Outcome] =
"Caregiver strain index" then "CSI" else if [Protocol Outcome] = "Psychological
distress/mood" then "PD" else if [Protocol Outcome] = "Carer generic health-related
quality of life" then "CHQOL" else if [Protocol Outcome] = "Stroke-specific Patient-
Reported Outcome Measures" then "SQOL" else "?"),
    #"Merged Queries" = Table.NestedJoin(#"Making short outcome", {"Study"},
BiasInd, {"Study"}, "Critical appraisal no edits", JoinKind.LeftOuter),
```

#"Expanded Critical appraisal no edits" = Table.ExpandTableColumn(#"Merged Queries", "Critical appraisal no edits", {"Short population", "Short comparison"}, {"Short population", "Short comparison"}),
#"Removed Duplicates" = Table.Distinct(#"Expanded Critical appraisal no edits"),
#"Removed Columns" = Table.RemoveColumns(#"Removed Duplicates",{"Footnotes", "Study year", "GIV Mean", "GIV SE", "Experimental mean", "Experimental SD", "Experimental cases", "Experimental N", "Control mean", "Control SD", "Control cases", "Control N", "O-E", "Variance", "Weight", "Mean", "CI start", "CI end", "Outcome short", "Short population", "Short comparison"}),
#"Duplicated Column" = Table.DuplicateColumn(#"Removed Columns", "Protocol Outcome", "Protocol Outcome - Copy"),
#"Pivoted Column" = Table.Pivot(#"Duplicated Column", List.Distinct(#"Duplicated Column"[#"Protocol Outcome - Copy"]), "Protocol Outcome - Copy", "Study", List.Count),
#"Inserted Sum" = Table.AddColumn(#"Pivoted Column", "Number of studies", each List.Sum({[Proportion of patients requiring transfusion], [Thrombotic complications], [Duplicate outcome proportion of people requiring transfusion], [Serious adverse events], [#"All-cause mortality"], [Number of units of allogenic blood transfused], [Infection], [Length of stay], [Acute myocardial infarction], [Volume of allogenic blood transfused], [Total blood loss volume]}), type number),
#"Inserted Merged Column" = Table.AddColumn(#"Inserted Sum", "GRADE group", each Text.Combine({Text.From([Analysis group], "en-GB"), Text.From([Analysis number], "en-GB")}, "-"), type text),
#"Removed Columns1" = Table.RemoveColumns(#"Inserted Merged Column",{"Applicability", "Subgroup", "Proportion of patients requiring transfusion", "Thrombotic complications", "Duplicate outcome proportion of people requiring transfusion", "Serious adverse events", "All-cause mortality", "Number of units of allogenic blood transfused", "Infection", "Length of stay", "Acute myocardial infarction", "Volume of allogenic blood transfused", "Total blood loss volume"}),
#"Reordered Columns" = Table.ReorderColumns(#"Removed Columns1",{"Analysis group", "Analysis number", "GRADE group", "Protocol Outcome", "Scale", "Follow up time", "Number of studies"}),
#"Merged Queries1" = Table.NestedJoin(#"Reordered Columns", {"GRADE group"}, #"Risk of bias", {"GRADE code"}, "Risk of bias", JoinKind.LeftOuter),

```
    #"Expanded Risk of bias" = Table.ExpandTableColumn(#"Merged Queries1", "Risk of bias", {"RoB", "RoB reason"}, {"RoB", "RoB reason"}),
    #"Merged Queries3" = Table.NestedJoin(#"Expanded Risk of bias", {"GRADE group"}, #"Inconsistency, units", {"GRADE group"}, "Inconsistency", JoinKind.LeftOuter),
    #"Expanded Inconsistency" = Table.ExpandTableColumn(#"Merged Queries3", "Inconsistency", {"Importance", "Inconsistency", "Inconsistency reason", "Other considerations", "ROBINS-I used?", "Study type", "Units"}, {"Importance", "Inconsistency.1", "Inconsistency reason", "Other considerations", "ROBINS-I used?", "Study type", "Units"}),
    #"Renamed Columns2" = Table.RenameColumns(#"Expanded Inconsistency",{{"Inconsistency.1", "Inconsistency"}}),
    #"Merged Queries2" = Table.NestedJoin(#"Renamed Columns2", {"GRADE group"}, Indirectness, {"GRADE code"}, "Indirectness", JoinKind.LeftOuter),
    #"Expanded Indirectness" = Table.ExpandTableColumn(#"Merged Queries2", "Indirectness", {"Indirectness", "Indirectness reason"}, {"Indirectness", "Indirectness reason"}),
    #"Merged Queries4" = Table.NestedJoin(#"Expanded Indirectness", {"GRADE group"}, Imprecision, {"GRADE group"}, "Imprecision", JoinKind.LeftOuter),
    #"Expanded Imprecision" = Table.ExpandTableColumn(#"Merged Queries4", "Imprecision", {"Imprecision", "Imprecision reason", "Sample size"}, {"Imprecision", "Imprecision reason", "Sample size"}),
    #"Merged Queries5" = Table.NestedJoin(#"Expanded Imprecision", {"GRADE group"}, Timepoint, {"GRADE code"}, "Timepoint", JoinKind.LeftOuter),
    #"Expanded Timepoint" = Table.ExpandTableColumn(#"Merged Queries5", "Timepoint", {"Average timepoint (months)"}, {"Average timepoint (months)"}),
    #"Added Conditional Column1" = Table.AddColumn(#"Expanded Timepoint", "ROBINS-I adjustment", each if [Study type] = "Randomised trials" then 0 else if [#"ROBINS-I used?"] = true then 0 else 2),
    #"Calculating RoB relative number" = Table.AddColumn(#"Added Conditional Column1", "RoB number", each if [RoB] = "Not serious" then 0 else if [RoB] = "Serious" then 1 else if [RoB] = "Very serious" then 2 else "?"),
    #"Calculating indirectness relative number" = Table.AddColumn(#"Calculating RoB relative number", "Indirectness number", each if [Indirectness] = "Not serious" then 0
```

```
else if [Indirectness] = "Serious" then 1 else if [Indirectness] = "Very serious" then 2 else "?"),
    #"Calculating inconsistency number" = Table.AddColumn(#"Calculating indirectness relative number", "Inconsistency number", each if [Inconsistency] = "Not serious" then 0 else if [Inconsistency] = "Serious" then 1 else if [Inconsistency] = "Very serious" then 2 else "?"),
    #"Calculating imprecision relative number" = Table.AddColumn(#"Calculating inconsistency number", "Imprecision number", each if [Imprecision] = "Not serious" then 0 else if [Imprecision] = "Serious" then 1 else if [Imprecision] = "Very serious" then 2 else "?"),
    #"Changed Type3" = Table.TransformColumnTypes(#"Calculating imprecision relative number",{{"RoB number", Int64.Type}, {"Indirectness number", Int64.Type}, {"Inconsistency number", Int64.Type}, {"Imprecision number", Int64.Type}, {"ROBINS-I adjustment", Int64.Type}}),
    #"Inserted Sum1" = Table.AddColumn(#"Changed Type3", "GRADE number", each List.Sum({[#"ROBINS-I adjustment"], [RoB number], [Indirectness number], [Inconsistency number], [Imprecision number]}), Int64.Type),
    #"Added Conditional Column" = Table.AddColumn(#"Inserted Sum1", "GRADE", each if [GRADE number] = 0 then "High" else if [GRADE number] = 1 then "Moderate" else if [GRADE number] = 2 then "Low" else if [GRADE number] >= 3 then "Very low" else "?"),
    #"Inserted Merged Column1" = Table.AddColumn(#"Added Conditional Column", "Full reasons", each Text.Combine({[RoB reason], [Inconsistency reason], [Indirectness reason], [Imprecision reason]}, "
"), type text),
    #"Completing reasons" = Table.AddColumn(#"Inserted Merged Column1", "Reasons", each if [Full reasons] = "" then "No downgrading required" else [Full reasons]),
    #"Removed Columns5" = Table.RemoveColumns(#"Completing reasons",{"Full reasons"}),
    #"Analyst edits for scale column" = Table.AddColumn(#"Removed Columns5", "Outcome scale", each if [Scale] = "days" then "Not applicable" else if [Scale] = "mL" then "Not applicable" else if [Scale] = "units" then "Not applicable" else if [Scale] <>
```

```
null then [Scale] else if [Protocol Outcome] = "Mortality" then "Not applicable" else
"Not applicable"),
    #"Risk of bias short reason" = Table.AddColumn(#"Analyst edits for scale column",
"Short RoB reason", each if [RoB reason] <> null then "Risk of Bias" else null),
    #"Inconsistency short reason" = Table.AddColumn(#"Risk of bias short reason",
"Inconsistency short reason", each if [Inconsistency reason] <> null then
"Inconsistency" else null),
    #"Indirectness short reason" = Table.AddColumn(#"Inconsistency short reason",
"Indirectness short reason", each if [Indirectness reason] <> null then "Indirectness"
else null),
    #"Imprecision short reason" = Table.AddColumn(#"Indirectness short reason",
"Imprecision short reason", each if [Imprecision reason] <> null then "Imprecision"
else null),
    #"Combining short reasons" = Table.AddColumn(#"Imprecision short reason",
"Short reason", each Text.Combine({[Short RoB reason], [Inconsistency short
reason], [Indirectness short reason], [Imprecision short reason]}, ", "), type text),
    #"Removed Columns2" = Table.RemoveColumns(#"Combining short
reasons",{"Short RoB reason", "Inconsistency short reason", "Indirectness short
reason", "Imprecision short reason"}),
    #"Added Custom" = Table.AddColumn(#"Removed Columns2", "Scale: ", each
"Scale: "),
    #"Added Custom1" = Table.AddColumn(#"Added Custom", "Units: ", each "Units:
"),
    #"Added Custom2" = Table.AddColumn(#"Added Custom1", "Study types:", each
"Study types: "),
    #"Inserted Merged Column2" = Table.AddColumn(#"Added Custom2", "Scale:
value", each Text.Combine({[#"Scale: "], [Outcome scale]}, ""), type text),
    #"Inserted Merged Column3" = Table.AddColumn(#"Inserted Merged Column2",
"Units: value", each Text.Combine({[#"Units: "], [Units]}, ""), type text),
    #"Inserted Merged Column5" = Table.AddColumn(#"Inserted Merged Column3",
"Study type: value", each Text.Combine({[#"Study types:"], [Study type]}, ""), type
text),
    #"Removed Columns3" = Table.RemoveColumns(#"Inserted Merged
Column5",{"Scale: ", "Units: ", "Study types:"}),
```

```
    #"Inserted Merged Column4" = Table.AddColumn(#"Removed Columns3",
"Outcome table", each Text.Combine({[Protocol Outcome], [Follow up time]}, " at "),
type text),
    #"Lowercased Text" = Table.TransformColumns(#"Inserted Merged
Column4",{{"Scale: value", Text.Lower, type text}, {"Units: value", Text.Lower, type
text}, {"Study type: value", Text.Lower, type text}}),
    #"Replaced Value" = Table.ReplaceValue(#"Lowercased
Text","ml","mL",Replacer.ReplaceText,{"Units: value"}),
    #"Merged Columns" = Table.CombineColumns(#"Replaced Value",{"Outcome
table", "Study type: value", "Scale: value", "Units:
value"},Combiner.CombineTextByDelimiter(", ", QuoteStyle.None),"Outcome full
table"),
    #"Added Custom3" = Table.AddColumn(#"Merged Columns", "Risk of bias: ", each
"Risk of bias: "),
    #"Added Custom4" = Table.AddColumn(#"Added Custom3", "Indirectness: ", each
"Indirectness: "),
    #"Added Custom5" = Table.AddColumn(#"Added Custom4", "Inconsistency: ",
each "Inconsistency: "),
    #"Added Custom6" = Table.AddColumn(#"Added Custom5", "Imprecision: ", each
"Imprecision: "),
    #"Added Custom7" = Table.AddColumn(#"Added Custom6", "Other concerns: ",
each "Other considerations: "),
    #"Inserted Merged Column6" = Table.AddColumn(#"Added Custom7", "Risk of
bias: value", each Text.Combine({[#"Risk of bias: "], [RoB]}, ""), type text),
    #"Inserted Merged Column7" = Table.AddColumn(#"Inserted Merged Column6",
"Indirectness: value", each Text.Combine({[#"Indirectness: "], [Indirectness]}, ""), type
text),
    #"Inserted Merged Column8" = Table.AddColumn(#"Inserted Merged Column7",
"Inconsistency: value", each Text.Combine({[#"Inconsistency: "], [Inconsistency]}, ""),
type text),
    #"Inserted Merged Column9" = Table.AddColumn(#"Inserted Merged Column8",
"Imprecision: value", each Text.Combine({[#"Imprecision: "], [Imprecision]}, ""), type
text),
```

```
    #"Inserted Merged Column10" = Table.AddColumn(#"Inserted Merged Column9",
"Other considerations: value", each Text.Combine({[#"Other concerns: "], [Other
considerations]}, ""), type text),
    #"Inserted Merged Column11" = Table.AddColumn(#"Inserted Merged Column10",
"GRADE components", each Text.Combine({[#"Risk of bias: value"], [#"Indirectness:
value"], [#"Inconsistency: value"], [#"Imprecision: value"], [#"Other considerations:
value"]}, "
"), type text),
    #"Removed Columns4" = Table.RemoveColumns(#"Inserted Merged
Column11",{"Risk of bias: ", "Indirectness: ", "Inconsistency: ", "Imprecision: ", "Other
concerns: ", "Risk of bias: value", "Indirectness: value", "Inconsistency: value",
"Imprecision: value", "Other considerations: value"})
in
    #"Removed Columns4"
```

Appendix B Review [B] Power M query for GRADE automation

Data rows

```
let
    Source = Csv.Document(Web.Contents("source"),[Delimiter=","]),
    #"Promoted Headers" = Table.PromoteHeaders(Source,
[PromoteAllScalars=true]),
    #"Changed Type" = Table.TransformColumnTypes(#"Promoted
Headers",{{"Analysis group", Int64.Type}, {"Analysis number", Int64.Type}, {"Analysis
name", type text}, {"Subgroup", type any}, {"Applicability", type text}, {"Study", type
text}, {"Study year", Int64.Type}, {"GIV Mean", type number}, {"GIV SE", type
number}, {"Experimental mean", type number}, {"Experimental SD", type number},
{"Experimental cases", Int64.Type}, {"Experimental N", Int64.Type}, {"Control mean",
type number}, {"Control SD", type number}, {"Control cases", Int64.Type}, {"Control
N", Int64.Type}, {"O-E", type any}, {"Variance", type any}, {"Weight", type number},
{"Mean", type number}, {"CI start", type number}, {"CI end", type number},
{"Footnotes", type any}}),
```

```
    #"Splitting by "" at "" to separate follow up time" = Table.SplitColumn(#"Changed
Type", "Analysis name", Splitter.SplitTextByDelimiter(" at ", QuoteStyle.Csv),
{"Analysis name.1", "Analysis name.2"}),
    #"Changed Type1" = Table.TransformColumnTypes(#"Splitting by "" at "" to
separate follow up time",{{"Analysis name.1", type text}, {"Analysis name.2", type
text}}),
    #"Renamed Columns" = Table.RenameColumns(#"Changed Type1",{{"Analysis
name.2", "Follow up time"}}),
    #"Splitting by "" ("" to separate scale information" = Table.SplitColumn(#"Renamed
Columns", "Analysis name.1", Splitter.SplitTextByDelimiter(" (", QuoteStyle.Csv),
{"Analysis name.1.1", "Analysis name.1.2"}),
    #"Changed Type2" = Table.TransformColumnTypes(#"Splitting by "" ("" to
separate scale information",{{"Analysis name.1.1", type text}, {"Analysis name.1.2",
type text}}),
    #"Renamed Columns1" = Table.RenameColumns(#"Changed Type2",{{"Analysis
name.1.2", "Scale"}, {"Analysis name.1.1", "Protocol Outcome"}}),
    #"Removing closing bracket" = Table.ReplaceValue(#"Renamed
Columns1",")","",Replacer.ReplaceText,{"Scale"}),
    #"Making short outcome" = Table.AddColumn(#"Removing closing bracket",
"Outcome short", each if [Protocol Outcome] = "Thromboembolic events after
surgery" then "Thromboembolic events" else if [Protocol Outcome] = "All-cause
mortality" then "Mortality" else if [Protocol Outcome] = "Pulmonary embolism" then
"PE" else if [Protocol Outcome] = "Deep vein thrombosis" then "DVT" else if [Protocol
Outcome] = "Myocardial infarction" then "MI" else if [Protocol Outcome] = "Ischaemic
stroke" then "Stroke" else if [Protocol Outcome] = "Infection" then "Infection" else if
[Protocol Outcome] = "All-cause readmission" then "Readmission" else if [Protocol
Outcome] = "Seizures" then "Seizures" else if [Protocol Outcome] = "Reoperation"
then "Reoperation" else "?"),
    #"Merged Queries" = Table.NestedJoin(#"Making short outcome", {"Study ID"},
#"Study level data", {"Study ID"}, "BiasInd", JoinKind.LeftOuter),
    #"Expanded Critical appraisal no edits" = Table.ExpandTableColumn(#"Merged
Queries", "BiasInd", {"Short population", "Short comparison"}, {"Short population",
"Short comparison"}),
    #"Removed Duplicates" = Table.Distinct(#"Expanded Critical appraisal no edits"),
```

```
    #"Removed Columns" = Table.RemoveColumns(#"Removed
Duplicates",{"Footnotes"}),
    #"Inserted Merged Column" = Table.AddColumn(#"Removed Columns", "GRADE
number", each Text.Combine({Text.From([Analysis group], "en-GB"),
Text.From([Analysis number], "en-GB")}, "-"), type text),
    #"Filtered Rows" = Table.SelectRows(#"Inserted Merged Column", each ([GRADE
number] <> "5-1" and [GRADE number] <> "5-10" and [GRADE number] <> "5-11"
and [GRADE number] <> "5-12" and [GRADE number] <> "5-13" and [GRADE
number] <> "5-14" and [GRADE number] <> "5-2" and [GRADE number] <> "5-3"
and [GRADE number] <> "5-4" and [GRADE number] <> "5-5" and [GRADE number]
<> "5-6" and [GRADE number] <> "5-7" and [GRADE number] <> "5-8" and [GRADE
number] <> "5-9" and [GRADE number] <> "6-1" and [GRADE number] <> "6-10"
and [GRADE number] <> "6-11" and [GRADE number] <> "6-12" and [GRADE
number] <> "6-13" and [GRADE number] <> "6-14" and [GRADE number] <> "6-15"
and [GRADE number] <> "6-16" and [GRADE number] <> "6-17" and [GRADE
number] <> "6-18" and [GRADE number] <> "6-19" and [GRADE number] <> "6-2"
and [GRADE number] <> "6-20" and [GRADE number] <> "6-21" and [GRADE
number] <> "6-3" and [GRADE number] <> "6-4" and [GRADE number] <> "6-5" and
[GRADE number] <> "6-6" and [GRADE number] <> "6-7" and [GRADE number] <>
"6-8" and [GRADE number] <> "6-9")),
    #"Replaced Value" = Table.ReplaceValue(#"Filtered
Rows",null,0,Replacer.ReplaceValue,{"Mean", "CI start", "CI end"}),
    #"Replaced Value1" = Table.ReplaceValue(#"Replaced
Value",0,0.0001,Replacer.ReplaceValue,{"CI end"}),
    #"Risk intervention" = Table.AddColumn(#"Replaced Value1", "Risk Intervention",
each [Experimental cases] / [Experimental N], type number),
    #"Risk control" = Table.AddColumn(#"Risk intervention", "Risk control", each
[Control cases] / [Control N], type number),
    #"Risk ratio" = Table.AddColumn(#"Risk control", "Risk ratio", each [Risk
Intervention] / [Risk control], type number),
    #"1/a" = Table.AddColumn(#"Risk ratio", "1/a", each Number.Power([Experimental
cases], -1), Int64.Type),
    #"1/c" = Table.AddColumn(#"1/a", "1/c", each Number.Power([Control cases], -1),
Int64.Type),
```

```
    #"a+b" = Table.AddColumn(#"1/c", "a+b", each [Experimental cases] +
[Experimental N], Int64.Type),
    #"c+d" = Table.AddColumn(#"a+b", "c+d", each [Control cases] + [Control N],
Int64.Type),
    #"1/a+b" = Table.AddColumn(#"c+d", "1/a+b", each Number.Power([#"a+b"], -1),
type number),
    #"1/c+d" = Table.AddColumn(#"1/a+b", "1/c+d", each Number.Power([#"c+d"], -1),
type number),
    #"Inserted Natural Logarithm" = Table.AddColumn(#"1/c+d", "ln(RR)", each
Number.Ln([Risk ratio]), type number),
    #"Inserted Addition" = Table.AddColumn(#"Inserted Natural Logarithm", "1/a +
1/c", each [#"1/a"] + [#"1/c"], Int64.Type),
    #"Inserted Subtraction" = Table.AddColumn(#"Inserted Addition", "Subtraction",
each [#"1/a + 1/c"] - [#"1/a+b"], type number),
    #"Inserted Subtraction1" = Table.AddColumn(#"Inserted Subtraction",
"Subtraction.1", each [Subtraction] - [#"1/c+d"], type number),
    #"Inserted Square Root" = Table.AddColumn(#"Inserted Subtraction1", "Square
Root", each Number.Sqrt([Subtraction.1]), type number),
    #"1.96*SQRT" = Table.AddColumn(#"Inserted Square Root", "1.96*SQRT", each
[Square Root] * 1.96, type number),
    #"ln(RR)+1.96*SQRT" = Table.AddColumn(#"1.96*SQRT", "Addition", each
[#"ln(RR)"] + [#"1.96*SQRT"], type number),
    #"ln(RR)-1.96*SQRT" = Table.AddColumn(#"ln(RR)+1.96*SQRT", "Subtraction.2",
each [#"ln(RR)"] - [#"1.96*SQRT"], type number),
    #"97.5% CI" = Table.AddColumn(#"ln(RR)-1.96*SQRT", "97.5% CI", each
Number.Exp([Addition]), type number),
    #"2.5% CI" = Table.AddColumn(#"97.5% CI", "2.5% CI", each
Number.Exp([Subtraction.2]), type number),
    #"Removed Columns1" = Table.RemoveColumns(#"2.5% CI",{"1/a", "1/c", "a+b",
"c+d", "1/a+b", "1/c+d", "ln(RR)", "1/a + 1/c", "Subtraction", "Subtraction.1", "Square
Root", "1.96*SQRT", "Addition", "Subtraction.2", "Risk Intervention", "Risk control"}),
    #"Added Conditional Column" = Table.AddColumn(#"Removed Columns1", "Risk
ratio f", each if [Risk ratio] = null then null else if [Risk ratio] >= 0.0001 then [Risk
ratio] else null),
```

```
    #"Added Conditional Column1" = Table.AddColumn(#"Added Conditional Column", "2.5% CI f", each if [#"2.5% CI"] = null then null else if [#"2.5% CI"] >= 1E-05 then [#"2.5% CI"] else null),
    #"Added Conditional Column2" = Table.AddColumn(#"Added Conditional Column1", "97.5% CI f", each if [#"97.5% CI"] = null then null else if [#"97.5% CI"] >= 1E-05 then [#"97.5% CI"] else null),
    #"Filtered Rows1" = Table.SelectRows(#"Added Conditional Column2", each ([Scale] <> "risk difference")),
    #"Replaced Value2" = Table.ReplaceValue(#"Filtered Rows1","risk ratio",null,Replacer.ReplaceValue,{"Scale"}),
    #"Inserted Merged Column1" = Table.AddColumn(#"Replaced Value2", "Study ID-Outcome", each Text.Combine({[Study ID], [Protocol Outcome], [Scale]}, "-"), type text)
in
    #"Inserted Merged Column1"
```

Overall estimates

```
let
    Source = Csv.Document(Web.Contents("source"),[Delimiter=","]),
    #"Promoted Headers" = Table.PromoteHeaders(Source, [PromoteAllScalars=true]),
    #"Changed Type" = Table.TransformColumnTypes(#"Promoted Headers",{{"Analysis group", Int64.Type}, {"Analysis number", Int64.Type}, {"Analysis name", type text}, {"Analysis group name", type text}, {"Data source", type text}, {"Data source eligibility", type text}, {"Data type", type text}, {"Log-scale data", type logical}, {"Outcome", type text}, {"Intervention grouping", type text}, {"Experimental intervention", type any}, {"Control intervention", type any}, {"Subgroup by", type any}, {"Filter criteria", type any}, {"Experimental group label", type text}, {"Control group label", type text}, {"Statistical method", type text}, {"Effect measure", type text}, {"Unit of effect measure", type any}, {"Analysis model", type text}, {"Heterogeneity estimator", type any}, {"Tau² CI", type logical}, {"Subgroup estimates", type logical}, {"Overall estimates", type logical}, {"Test for subgroup differences", type logical}, {"Prediction interval", type logical}, {"Swap event and non-event", type logical}, {"CI method", type text}, {"CI/PI level", type number}, {"Experimental cases", Int64.Type},
```

{"Experimental N", Int64.Type}, {"Control cases", Int64.Type}, {"Control N", Int64.Type}, {"Mean", type number}, {"CI start", type number}, {"CI end", type number}, {"PI start", type any}, {"PI end", type any}, {"Heterogeneity Tau²", type any}, {"Tau² CI start", type any}, {"Tau² CI end", type any}, {"Heterogeneity Chi²", type number}, {"Heterogeneity df", Int64.Type}, {"Heterogeneity P", type number}, {"Heterogeneity I²", type number}, {"Effect Z", type number}, {"Effect P", type number}, {"Subgroup Chi²", type any}, {"Subgroup df", type any}, {"Subgroup P", type any}, {"Subgroup I²", type any}, {"ID", type text}}),

#"Removed Columns" = Table.RemoveColumns(#"Changed Type",{"PI start", "PI end", "Heterogeneity Tau²", "Tau² CI start", "Tau² CI end", "Heterogeneity Chi²", "Heterogeneity df", "Heterogeneity P", "Effect Z", "Effect P", "Subgroup Chi²", "Subgroup df", "Subgroup P", "ID", "CI method", "Test for subgroup differences", "Prediction interval", "Swap event and non-event", "Tau² CI", "Heterogeneity estimator", "Experimental intervention", "Control intervention"}),

#"Inserted Merged Column" = Table.AddColumn(#"Removed Columns", "GRADE group", each Text.Combine({Text.From([Analysis group], "en-GB"), Text.From([Analysis number], "en-GB")}, "-"), type text),

#"Analyst editable control risk rate (proportion)" = Table.AddColumn(#"Inserted Merged Column", "Control risk rate (proportion)", each if [Outcome] = "a" then 0.2 else null),

#"Changed Type6" = Table.TransformColumnTypes(#"Analyst editable control risk rate (proportion)",{{"Control risk rate (proportion)", type number}}),

#"Adjusted for editable control n calculation" = Table.AddColumn(#"Changed Type6", "Adjusted control N", each [#"Control risk rate (proportion)"] * [Control N], type number),

#"Changed Type8" = Table.TransformColumnTypes(#"Adjusted for editable control n calculation",{{"Adjusted control N", Int64.Type}}),

#"Merged Queries" = Table.NestedJoin(#"Changed Type8", {"GRADE group"}, #"Median baseline rate", {"GRADE number"}, "Median baseline rate", JoinKind.LeftOuter),

#"Expanded Median baseline rate" = Table.ExpandTableColumn(#"Merged Queries", "Median baseline rate", {"Median control group rate"}, {"Median control group rate"}),

```
    #"Adjusted for median control group rate" = Table.AddColumn(#"Expanded
Median baseline rate", "Median control n", each [Median control group rate] * [Control
N], type number),
    #"Changed Type7" = Table.TransformColumnTypes(#"Adjusted for median control
group rate",{{"Median control n", Int64.Type}}),
    #"Renamed Columns4" = Table.RenameColumns(#"Changed Type7",{{"Control
N", "True Control N"}}),
    #"Analyst editable choice of control group N" = Table.AddColumn(#"Renamed
Columns4", "Control N", each if [Outcome] = "a" then [Adjusted control N] else if
[Outcome] = "b" then [Median control n] else if [Outcome] = "c" then [True Control N]
else [True Control N]),
    #"Calculating non-events 1" = Table.AddColumn(#"Analyst editable choice of
control group N", "Non-events 1", each [Experimental N] - [Experimental cases],
Int64.Type),
    #"Calculating non-events 2" = Table.AddColumn(#"Calculating non-events 1",
"Non-events 2", each [Control N] - [Control cases], Int64.Type),
    #"Calculating control rate" = Table.AddColumn(#"Calculating non-events 2",
"Baseline control rate", each ([Control cases] / [Control N]) * 1000, Int64.Type),
    #"Inserted Multiplication6" = Table.AddColumn(#"Calculating control rate",
"Baseline control rate/10,000", each [Baseline control rate] * 10, type number),
    #"Changed Type9" = Table.TransformColumnTypes(#"Inserted
Multiplication6",{{"Baseline control rate", Int64.Type}, {"Baseline control rate/10,000",
Int64.Type}}),
    #"per 1000 people" = Table.AddColumn(#"Changed Type9", "per 1000 people",
each " per 10,000 people"),
    #"Making control group rate" =
Table.CombineColumns(Table.TransformColumnTypes(#"per 1000 people",
{{"Baseline control rate/10,000", type text}}, "en-GB"),{"Baseline control rate/10,000",
"per 1000 people"},Combiner.CombineTextByDelimiter("", QuoteStyle.None),"Control
group rate"),
    #"Replaced Value" = Table.ReplaceValue(#"Making control group rate"," per
10,000 people","Not applicable",Replacer.ReplaceValue,{"Control group rate"}),
    #"Calculating risk1" = Table.AddColumn(#"Replaced Value", "Risk1", each
[Experimental cases] / [Experimental N], type number),
```

```
    #"Calculating risk2" = Table.AddColumn(#"Calculating risk1", "Risk2", each
[Control cases] / [Control N], type number),
    #"Renamed Columns" = Table.RenameColumns(#"Calculating
risk2",{{"Experimental cases", "Number of events 1"}, {"Experimental N", "n1"},
{"Control cases", "Number of events 2"}, {"Control N", "n2"}}),
    #"Calculating risk difference" = Table.AddColumn(#"Renamed Columns", "Risk
difference", each [Risk1] - [Risk2], type number),
    #"Inserted Multiplication1" = Table.AddColumn(#"Calculating risk difference",
"Event1xNonevent1", each [Number of events 1] * [#"Non-events 1"], Int64.Type),
    #"Inserted Multiplication2" = Table.AddColumn(#"Inserted Multiplication1",
"EventxNonevent2", each [Number of events 2] * [#"Non-events 2"], Int64.Type),
    #"Inserted Cube" = Table.AddColumn(#"Inserted Multiplication2", "n1^3", each
Number.Power([#"Non-events 1"], 3), Int64.Type),
    #"Inserted Cube1" = Table.AddColumn(#"Inserted Cube", "n2^3", each
Number.Power([n2], 3), Int64.Type),
    #"Inserted Division3" = Table.AddColumn(#"Inserted Cube1",
"eventxnonevent1/n1^3", each [Event1xNonevent1] / [#"n1^3"], type number),
    #"Inserted Division4" = Table.AddColumn(#"Inserted Division3", "Division", each
[EventxNonevent2] / [#"n2^3"], type number),
    #"Renamed Columns1" = Table.RenameColumns(#"Inserted
Division4",{{"Division", "eventxnonevent2/n2^3"}}),
    #"Inserted Addition2" = Table.AddColumn(#"Renamed Columns1",
"eventxnonevent1/n1^3 + eventxnonevent2/n2^3", each [#"eventxnonevent1/n1^3"] +
[#"eventxnonevent2/n2^3"], type number),
    #"Inserted Square Root1" = Table.AddColumn(#"Inserted Addition2", "SE(RD)",
each Number.Sqrt([#"eventxnonevent1/n1^3 + eventxnonevent2/n2^3"]), type
number),
    #"Inserted Multiplication3" = Table.AddColumn(#"Inserted Square Root1",
"1.96*SE", each [#"SE(RD)"] * 1.96, type number),
    #"Inserted Subtraction4" = Table.AddColumn(#"Inserted Multiplication3", "RD 2.5%
CI", each [Risk difference] - [#"1.96*SE"], type number),
    #"Inserted Addition3" = Table.AddColumn(#"Inserted Subtraction4", "RD 97.5%
CI", each [Risk difference] + [#"1.96*SE"], type number),
```

```
    #"Removed Columns3" = Table.RemoveColumns(#"Inserted
Addition3",{"Event1xNonevent1", "EventxNonevent2", "n1^3", "n2^3",
"eventxnonevent1/n1^3", "eventxnonevent2/n2^3", "eventxnonevent1/n1^3 +
eventxnonevent2/n2^3", "SE(RD)", "1.96*SE", "Non-events 1", "Non-events 2",
"Risk1"}),
    #"Calculating events/1000 people" = Table.AddColumn(#"Removed Columns3",
"Events/1000 RD", each [Risk difference] * 1000, type number),
    #"Calculating 2.5% CI" = Table.AddColumn(#"Calculating events/1000 people",
"Events/1000 2.5% CI RD", each [#"RD 2.5% CI"] * 1000, type number),
    #"Calculating 97.5% CI" = Table.AddColumn(#"Calculating 2.5% CI", "Events/1000
97.5% CI RD", each [#"RD 97.5% CI"] * 1000, type number),
    #"Calculating events/1000 people mean" = Table.AddColumn(#"Calculating 97.5%
CI", "Events/1000 mean", each [Mean] * 1000, type number),
    #"Calculating 2.5% CI mean" = Table.AddColumn(#"Calculating events/1000
people mean", "Events/1000 2.5% CI mean", each [#"CI start"] * 1000, type number),
    #"Calculating 97.5% CI mean" = Table.AddColumn(#"Calculating 2.5% CI mean",
"Events/1000 97.5% CI mean", each [#"CI end"] * 1000, type number),
    #"Calculating events/10000 people" = Table.AddColumn(#"Calculating 97.5% CI
mean", "Events/10000 RD", each [Risk difference] * 10000, type number),
    #"Calculating 2.5% CI 10000" = Table.AddColumn(#"Calculating events/10000
people", "Events/10000 2.5% CI RD", each [#"RD 2.5% CI"] * 10000, type number),
    #"Calculating 97.5% CI 10000" = Table.AddColumn(#"Calculating 2.5% CI 10000",
"Events/10000 97.5% CI RD", each [#"RD 97.5% CI"] * 10000, type number),
    #"Calculating events/10000 people mean" = Table.AddColumn(#"Calculating
97.5% CI 10000", "Events/10000 mean", each [Mean] * 10000, type number),
    #"Calculating 2.5% CI mean 10000" = Table.AddColumn(#"Calculating
events/10000 people mean", "Events/10000 2.5% CI mean", each [#"CI start"] *
10000, type number),
    #"Calculating 97.5% CI mean 10000" = Table.AddColumn(#"Calculating 2.5% CI
mean 10000", "Events/10000 97.5% CI mean", each [#"CI end"] * 10000, type
number),
    #"Changed Type1" = Table.TransformColumnTypes(#"Calculating 97.5% CI mean
10000",{{"Events/1000 97.5% CI RD", Int64.Type}, {"Events/1000 2.5% CI RD",
Int64.Type}, {"Events/1000 RD", Int64.Type}, {"Events/1000 97.5% CI mean",
```

Int64.Type}, {"Events/1000 2.5% CI mean", Int64.Type}, {"Events/1000 mean", Int64.Type}, {"Events/10000 RD", Int64.Type}, {"Events/10000 2.5% CI RD", Int64.Type}, {"Events/10000 97.5% CI RD", Int64.Type}, {"Events/10000 mean", Int64.Type}, {"Events/10000 2.5% CI mean", Int64.Type}, {"Events/10000 97.5% CI mean", Int64.Type}}),

#"Analyst choice events/1000 method" = Table.AddColumn(#"Changed Type1", "Events/1000", each if [Effect measure] = "Risk Difference" then [#"Events/1000 mean"] else [#"Events/1000 RD"]),

#"Analyst choice 2.5% CI" = Table.AddColumn(#"Analyst choice events/1000 method", "Events/1000 2.5% CI", each if [Effect measure] = "Risk Difference" then [#"Events/1000 2.5% CI mean"] else [#"Events/1000 2.5% CI RD"]),

#"Analyst choice 97.5% CI" = Table.AddColumn(#"Analyst choice 2.5% CI", "Events/1000 97.5% CI", each if [Effect measure] = "Risk Difference" then [#"Events/1000 97.5% CI mean"] else [#"Events/1000 97.5% CI RD"]),

#"Analyst choice events/10000 method" = Table.AddColumn(#"Analyst choice 97.5% CI", "Events/10000", each if [Effect measure] = "Risk Difference" then [#"Events/10000 mean"] else [#"Events/10000 RD"]),

#"Analyst choice 2.5% CI 10000" = Table.AddColumn(#"Analyst choice events/10000 method", "Events/10000 2.5% CI", each if [Effect measure] = "Risk Difference" then [#"Events/10000 2.5% CI mean"] else [#"Events/10000 2.5% CI RD"]),

#"Analyst choice 97.5% CI 10000" = Table.AddColumn(#"Analyst choice 2.5% CI 10000", "Events/10000 97.5% CI", each if [Effect measure] = "Risk Difference" then [#"Events/10000 97.5% CI mean"] else [#"Events/10000 97.5% CI RD"]),

#"Extracting RD direction of effect" = Table.AddColumn(#"Analyst choice 97.5% CI 10000", "Sign RD", each Number.Sign([#"Events/1000"]), Int64.Type),

#"Extracting 2.5% CI direction of effect" = Table.AddColumn(#"Extracting RD direction of effect", "Sign RD 2.5%", each Number.Sign([#"Events/1000 2.5% CI"]), Int64.Type),

#"Extracting 97.5% CI direction of effect" = Table.AddColumn(#"Extracting 2.5% CI direction of effect", "Sign RD 97.5%", each Number.Sign([#"Events/1000 97.5% CI"]), Int64.Type),

#"Determining words after RD" = Table.AddColumn(#"Extracting 97.5% CI direction of effect", "RD wording", each if [Sign RD] = 1 then " more events per

```
10,000 people, " else if [Sign RD] = -1 then " fewer events per 10,000 people, " else if [Sign RD] = 0 then " fewer events per 10,000 people, " else if [Sign RD] = null then null else "?"),
    #"Determining words after RD 2.5% CI" = Table.AddColumn(#"Determining words after RD", "RD 2.5% CI wording", each if [#"Sign RD 2.5%"] = 1 then " more to " else if [#"Sign RD 2.5%"] = -1 then " fewer to " else if [#"Sign RD 2.5%"] = 0 then " fewer to " else if [#"RD 2.5% CI"] = null then null else "?"),
    #"Determining words after RD 97.5% CI" = Table.AddColumn(#"Determining words after RD 2.5% CI", "RD 97.5% CI wording", each if [#"Sign RD 97.5%"] = 1 then " more" else if [#"Sign RD 97.5%"] = -1 then " fewer" else if [#"Sign RD 97.5%"] = 0 then " fewer" else if [#"Sign RD 97.5%"] = null then null else "?"),
    #"Maintain RD direction" = Table.DuplicateColumn(#"Determining words after RD 97.5% CI", "Events/1000", "Events/1000 direction"),
    #"Maintain RD 2.5% direction" = Table.DuplicateColumn(#"Maintain RD direction", "Events/1000 2.5% CI", "Events/1000 2.5% CI direction"),
    #"Maintain RD 97.5% direction" = Table.DuplicateColumn(#"Maintain RD 2.5% direction", "Events/1000 97.5% CI", "Events/1000 97.5% CI direction"),
    #"Calculated Absolute Value" = Table.TransformColumns(#"Maintain RD 97.5% direction",{{"Events/1000", Number.Abs, Int64.Type}, {"Events/1000 2.5% CI", Number.Abs, Int64.Type}, {"Events/1000 97.5% CI", Number.Abs, Int64.Type}}),
    #"Inserted Absolute Value" = Table.AddColumn(#"Calculated Absolute Value", "Absolute Value", each Number.Abs([#"Events/10000"]), type number),
    #"Inserted Absolute Value1" = Table.AddColumn(#"Inserted Absolute Value", "Absolute Value.1", each Number.Abs([#"Events/10000 2.5% CI"]), type number),
    #"Inserted Absolute Value2" = Table.AddColumn(#"Inserted Absolute Value1", "Absolute Value.2", each Number.Abs([#"Events/10000 97.5% CI"]), type number),
    #"Renamed Columns9" = Table.RenameColumns(#"Inserted Absolute Value2",{{"Absolute Value.2", "Events/10000 97.5% CI a"}}),
    #"Renamed Columns8" = Table.RenameColumns(#"Renamed Columns9",{{"Absolute Value.1", "Events/10000 2.5% CI a"}}),
    #"Renamed Columns7" = Table.RenameColumns(#"Renamed Columns8",{{"Absolute Value", "Events/10000 a"}}),
    #"Combine for written risk difference value" = Table.AddColumn(#"Renamed Columns7", "Risk difference effect", each Text.Combine({Text.From([#"Events/10000
```

```
a"], "en-GB"), [RD wording], Text.From([#"Events/10000 2.5% CI a"], "en-GB"), [#"RD 2.5% CI wording"], Text.From([#"Events/10000 97.5% CI a"], "en-GB"), [#"RD 97.5% CI wording"]}, ""), type text),
    #"Added Conditional Column" = Table.AddColumn(#"Combine for written risk difference value", "Absolute effect", each if [Risk difference effect] <> "" then [Risk difference effect] else if [Effect measure] = "Hazard Ratio" then "Not applicable" else if [Effect measure] = "Risk Ratio" then "Not applicable" else if [Effect measure] = "Odds Ratio" then "Not applicable" else "Not applicable"),
    #"Replaced Value1" = Table.ReplaceValue(#"Added Conditional Column","1 more events","1 more event",Replacer.ReplaceText,{"Absolute effect"}),
    #"Replaced Value2" = Table.ReplaceValue(#"Replaced Value1","1 fewer events","1 fewer event",Replacer.ReplaceText,{"Absolute effect"}),
    #"Inserted Multiplication" = Table.AddColumn(#"Replaced Value2", "Number of events in control group/10000 people", each [Risk2] * 10000, type number),
    #"Risk2 * 1000" = Table.TransformColumns(#"Inserted Multiplication", {{"Risk2", each _ * 1000, type number}}),
    #"Renamed Columns2" = Table.RenameColumns(#"Risk2 * 1000",{{"Risk2", "Number of events in control group/1000 people"}}),
    #"Changed Type2" = Table.TransformColumnTypes(#"Renamed Columns2",{{"Number of events in control group/1000 people", Int64.Type}, {"Number of events in control group/10000 people", Int64.Type}}),
    #"Removed Columns2" = Table.RemoveColumns(#"Changed Type2",{"Sign RD", "Sign RD 2.5%", "Sign RD 97.5%"}),
    #"Total events/1000" = Table.AddColumn(#"Removed Columns2", "Total events after treatment", each [#"Number of events in control group/1000 people"] + [#"Events/1000 direction"], Int64.Type),
    #"Total events/10000" = Table.AddColumn(#"Total events/1000", "Total events after treatment/10000", each [#"Number of events in control group/10000 people"] + [#"Events/10000"], type number),
    #"Determining smallest number of events/1000" = Table.AddColumn(#"Total events/10000", "Smallest number of events/1000", each if [Total events after treatment] = null then null else if [Total events after treatment] < [#"Number of events in control group/1000 people"] then [Total events after treatment] else [#"Number of events in control group/1000 people"]),
```

#"Smallest number of events/10000" = Table.AddColumn(#"Determining smallest number of events/1000", "Smallest number of events/10000", each if [#"Total events after treatment/10000"] = null then null else if [#"Total events after treatment/10000"] < [#"Number of events in control group/10000 people"] then [#"Total events after treatment/10000"] else [#"Number of events in control group/10000 people"]),

#"Changed Type4" = Table.TransformColumnTypes(#"Smallest number of events/10000",{{"Smallest number of events/1000", Int64.Type}, {"Smallest number of events/10000", Int64.Type}, {"Total events after treatment/10000", Int64.Type}}),

#"1000" = Table.AddColumn(#"Changed Type4", "1000", each "1000"),

#"10000" = Table.AddColumn(#"1000", "10000", each "10000"),

#"Additional events from treatment/1000" = Table.AddColumn(#"10000", "Additional events from treatment", each [Total events after treatment] - [#"Smallest number of events/1000"], type number),

#"Additional events from treatment/10000" = Table.AddColumn(#"Additional events from treatment/1000", "Additional events from treatment/10000", each [#"Total events after treatment/10000"] - [#"Smallest number of events/10000"], Int64.Type),

#"Events reduced from treatment/1000" = Table.AddColumn(#"Additional events from treatment/10000", "Events reduced from treatment", each [#"Number of events in control group/1000 people"] - [#"Smallest number of events/1000"], type number),

#"Events reduced from treatment/10000" = Table.AddColumn(#"Events reduced from treatment/1000", "Events reduced from treatment/10000", each [#"Number of events in control group/10000 people"] - [#"Smallest number of events/10000"], Int64.Type),

#"Changed Type3" = Table.TransformColumnTypes(#"Events reduced from treatment/10000",{{"1000", Int64.Type}, {"Smallest number of events/1000", Int64.Type}, {"Additional events from treatment", Int64.Type}, {"Events reduced from treatment", Int64.Type}, {"10000", Int64.Type}}),

#"Determining largest number of events/1000" = Table.AddColumn(#"Changed Type3", "Largest number of events/1000", each if [#"Number of events in control group/1000 people"] = null then null else if [#"Number of events in control group/1000 people"] >= [Total events after treatment] then [#"Number of events in control group/1000 people"] else [Total events after treatment]),

#"Largest number of events/10000" = Table.AddColumn(#"Determining largest number of events/1000", "Largest number of events/10000", each if [#"Number of

```
events in control group/10000 people"] = null then null else if [#"Number of events in control group/10000 people"] >= [#"Total events after treatment/10000"] then [#"Number of events in control group/10000 people"] else [#"Total events after treatment/10000"]),
    #"Changed Type5" = Table.TransformColumnTypes(#"Largest number of events/10000",{{"Largest number of events/1000", Int64.Type}, {"Largest number of events/10000", Int64.Type}}),
    #"People without event/1000" = Table.AddColumn(#"Changed Type5", "People without event", each [1000] - [#"Largest number of events/1000"], Int64.Type),
    #"People without event/10000" = Table.AddColumn(#"People without event/1000", "People without event/10000", each [10000] - [#"Largest number of events/10000"], Int64.Type),
    #"Renamed Columns3" = Table.RenameColumns(#"People without event/10000",{{"Smallest number of events/1000", "People with events"}, {"Events reduced from treatment", "Fewer events after treatment"}, {"Smallest number of events/10000", "People with events/10000"}, {"Events reduced from treatment/10000", "Fewer events after treatment/10000"}}),
    #"Added Custom1" = Table.AddColumn(#"Renamed Columns3", "|", each "|"),
    #"Filtered Rows" = Table.SelectRows(#"Added Custom1", each ([Analysis group] <> 5 and [Analysis group] <> 6)),
    #"Added Custom2" = Table.AddColumn(#"Filtered Rows", "(", each "("),
    #"Added Custom3" = Table.AddColumn(#"Added Custom2", ")", each ")"),
    #"Rounded Off" = Table.TransformColumns(#"Added Custom3",{{"Mean", each Number.Round(_, 5), type number}, {"CI start", each Number.Round(_, 5), type number}, {"CI end", each Number.Round(_, 5), type number}}),
    #"Inserted Merged Column1" = Table.AddColumn(#"Rounded Off", "Effect size", each Text.Combine({Text.From([Mean], "en-GB"), [#"("]}, " "), type text),
    #"Inserted Merged Column2" = Table.AddColumn(#"Inserted Merged Column1", "Effect Size.1", each Text.Combine({[Effect size], Text.From([CI start], "en-GB")}, ""), type text),
    #"Inserted Merged Column3" = Table.AddColumn(#"Inserted Merged Column2", "Effect size.2", each Text.Combine({[Effect Size.1], Text.From([CI end], "en-GB")}, ", "), type text),
```

```
    #"Inserted Merged Column4" = Table.AddColumn(#"Inserted Merged Column3",
"Effect size.3", each Text.Combine({[Effect size.2], [#")"]}, ""), type text),
    #"Removed Columns1" = Table.RemoveColumns(#"Inserted Merged
Column4",{"(", ")", "Effect size", "Effect Size.1", "Effect size.2"}),
    #"Renamed Columns5" = Table.RenameColumns(#"Removed Columns1",{{"Effect
size.3", "Effect size"}}),
    #"Duplicated Column" = Table.DuplicateColumn(#"Renamed Columns5", "Analysis
name", "Analysis name - Copy"),
    #"Split Column by Delimiter" = Table.SplitColumn(#"Duplicated Column", "Analysis
name - Copy", Splitter.SplitTextByDelimiter("(", QuoteStyle.Csv), {"Analysis name -
Copy.1", "Analysis name - Copy.2"}),
    #"Changed Type10" = Table.TransformColumnTypes(#"Split Column by
Delimiter",{{"Effect T", type text}, {"Analysis name - Copy.1", type text}, {"Analysis
name - Copy.2", type text}}),
    #"Removed Columns4" = Table.RemoveColumns(#"Changed Type10",{"Analysis
name - Copy.1"}),
    #"Extracted Text Before Delimiter" = Table.TransformColumns(#"Removed
Columns4", {{"Analysis name - Copy.2", each Text.BeforeDelimiter(_, ")"), type
text}}),
    #"Renamed Columns6" = Table.RenameColumns(#"Extracted Text Before
Delimiter",{{"Analysis name - Copy.2", "Scale"}}),
    #"Filtered Rows1" = Table.SelectRows(#"Renamed Columns6", each ([Scale] <>
"risk difference")),
    #"Replaced Value3" = Table.ReplaceValue(#"Filtered Rows1","Pregnancy -
","",Replacer.ReplaceText,{"Analysis group name"}),
    #"Replaced Value4" = Table.ReplaceValue(#"Replaced Value3","Not pregnant -
","",Replacer.ReplaceText,{"Analysis group name"}),
    #"Added Conditional Column1" = Table.AddColumn(#"Replaced Value4", "MID
multiplier", each if [Outcome] = "All-cause mortality" then 10 else if [Outcome] =
"Thromboembolic events after surgery" then 0.2 else 1),
    #"Inserted Multiplication4" = Table.AddColumn(#"Added Conditional Column1",
"Multiplication", each [Median control group rate] * [MID multiplier] * 1000, type
number),
```

```
    #"Rounded Up" = Table.TransformColumns(#"Inserted
Multiplication4",{{"Multiplication", Number.RoundUp, Int64.Type}}),
    #"Added Conditional Column2" = Table.AddColumn(#"Rounded Up", "MID", each if
[Outcome] = "All-cause mortality" then 10 else [Multiplication]),
    #"Inserted Multiplication5" = Table.AddColumn(#"Added Conditional Column2",
"Negative MID", each [MID] * -1, type number),
    #"Added Conditional Column3" = Table.AddColumn(#"Inserted Multiplication5",
"Positive MID side", each if [#"Events/10000"] = null then null else if
[#"Events/10000"] > [MID] then 1 else if [#"Events/10000"] = [MID] then 0 else if
[#"Events/10000"] < [MID] then -1 else null),
    #"Added Conditional Column4" = Table.AddColumn(#"Added Conditional
Column3", "Negative MID side", each if [#"Events/1000 direction"] = null then null
else if [#"Events/10000"] < [Negative MID] then -1 else if [#"Events/10000"] =
[Negative MID] then 0 else if [#"Events/10000"] > [Negative MID] then 1 else null),
    #"Inserted Merged Column5" = Table.AddColumn(#"Added Conditional Column4",
"MID code", each Text.Combine({Text.From([Positive MID side], "en-GB"),
Text.From([Negative MID side], "en-GB")}, "-"), type text),
    #"Replaced Value5" = Table.ReplaceValue(#"Inserted Merged
Column5","",null,Replacer.ReplaceValue,{"MID code"}),
    #"Added Conditional Column5" = Table.AddColumn(#"Replaced Value5", "Clinical
importance", each if [MID code] = "-1--1" then "Clinically important benefit" else if
[MID code] = "-1-0" then "Clinically important benefit" else if [MID code] = "1-1" then
"Clinically important harm" else if [MID code] = "-1-1" then "No clinically important
difference" else if [MID code] = "0-1" then "Clinically important harm" else null),
    #"Inserted Merged Column6" = Table.AddColumn(#"Added Conditional Column5",
"Absolute effect (clinical importance)", each Text.Combine({[Absolute effect], [Clinical
importance]}, "
"), type text),
    #"MID absolute" = Table.AddColumn(#"Inserted Merged Column6", "MID
absolute", each [MID], type number),
    #"Added Prefix" = Table.TransformColumns(#"MID absolute", {{"MID absolute",
each "MID (clinical importance) = " & Text.From(_, "en-GB"), type text}}),
    #"Added Suffix" = Table.TransformColumns(#"Added Prefix", {{"MID absolute",
each _ & " events per 10,000 people", type text}})
```

```
FINAL

in
    #"Added Suffix"
```

Risk of bias

```
let
    Source = Csv.Document(Web.Contents("source"),[Delimiter=","]),
    #"Promoted Headers" = Table.PromoteHeaders(Source,
[PromoteAllScalars=true]),
    #"Changed Type" = Table.TransformColumnTypes(#"Promoted
Headers",{{"Analysis group", Int64.Type}, {"Analysis number", Int64.Type}, {"Analysis
name", type text}, {"Subgroup", type any}, {"Applicability", type text}, {"Study", type
text}, {"Study year", Int64.Type}, {"GIV Mean", type number}, {"GIV SE", type
number}, {"Experimental mean", type number}, {"Experimental SD", type number},
{"Experimental cases", Int64.Type}, {"Experimental N", Int64.Type}, {"Control mean",
type number}, {"Control SD", type number}, {"Control cases", Int64.Type}, {"Control
N", Int64.Type}, {"O-E", type any}, {"Variance", type any}, {"Weight", type number},
{"Mean", type number}, {"CI start", type number}, {"CI end", type number},
{"Footnotes", type any}}),
    #"Splitting by "" at "" to separate follow up time" = Table.SplitColumn(#"Changed
Type", "Analysis name", Splitter.SplitTextByDelimiter(" at ", QuoteStyle.Csv),
{"Analysis name.1", "Analysis name.2"}),
    #"Changed Type1" = Table.TransformColumnTypes(#"Splitting by "" at "" to
separate follow up time",{{"Analysis name.1", type text}, {"Analysis name.2", type
text}}),
    #"Renamed Columns" = Table.RenameColumns(#"Changed Type1",{{"Analysis
name.2", "Follow up time"}}),
    #"Splitting by "" ("" to separate scale information" = Table.SplitColumn(#"Renamed
Columns", "Analysis name.1", Splitter.SplitTextByDelimiter(" (", QuoteStyle.Csv),
{"Analysis name.1.1", "Analysis name.1.2"}),
    #"Changed Type2" = Table.TransformColumnTypes(#"Splitting by "" ("" to
separate scale information",{{"Analysis name.1.1", type text}, {"Analysis name.1.2",
type text}}),
    #"Renamed Columns1" = Table.RenameColumns(#"Changed Type2",{{"Analysis
name.1.2", "Scale"}, {"Analysis name.1.1", "Protocol Outcome"}}),
```

```
    #"Removing closing bracket" = Table.ReplaceValue(#"Renamed
Columns1",")","",Replacer.ReplaceText,{"Scale"}),
    #"Making short outcome" = Table.AddColumn(#"Removing closing bracket",
"Outcome short", each if [Protocol Outcome] = "Thromboembolic events after
surgery" then "Thromboembolic events" else if [Protocol Outcome] = "All-cause
mortality" then "Mortality" else if [Protocol Outcome] = "Pulmonary embolism" then
"PE" else if [Protocol Outcome] = "Deep vein thrombosis" then "DVT" else if [Protocol
Outcome] = "Myocardial infarction" then "MI" else if [Protocol Outcome] = "Ischaemic
stroke" then "Stroke" else if [Protocol Outcome] = "Infection" then "Infection" else if
[Protocol Outcome] = "All-cause readmission" then "Readmission" else if [Protocol
Outcome] = "Seizures" then "Seizures" else if [Protocol Outcome] = "Reoperation"
then "Reoperation" else "?"),
    #"Replaced Value2" = Table.ReplaceValue(#"Making short outcome","risk
ratio",null,Replacer.ReplaceValue,{"Scale"}),
    #"Filtered Rows1" = Table.SelectRows(#"Replaced Value2", each ([Scale] <> "risk
difference")),
    #"Inserted Merged Column1" = Table.AddColumn(#"Filtered Rows1", "Study ID-
Outcome-Scale", each Text.Combine({[Study ID], [Protocol Outcome], [Scale]}, "-"),
type text),
    #"Merged Queries" = Table.NestedJoin(#"Inserted Merged Column1", {"Study ID"},
#"Study level data", {"Study ID"}, "BiasInd", JoinKind.LeftOuter),
    #"Expanded Critical appraisal no edits" = Table.ExpandTableColumn(#"Merged
Queries", "BiasInd", {"Follow-up", "Short population", "Short comparison"}, {"Follow-
up", "Short population", "Short comparison"}),
    #"Renamed Columns2" = Table.RenameColumns(#"Expanded Critical appraisal
no edits",{{"Follow-up", "Timepoint (days)"}}),
    #"Merged Queries1" = Table.NestedJoin(#"Renamed Columns2", {"Study ID-
Outcome-Scale"}, BiasInd, {"Study ID-Outcome-Scale"}, "BiasInd",
JoinKind.LeftOuter),
    #"Expanded BiasInd" = Table.ExpandTableColumn(#"Merged Queries1",
"BiasInd", {"Overall risk of bias", "Overall directness"}, {"Overall risk of bias", "Overall
directness"}),
    #"Removed Duplicates" = Table.Distinct(#"Expanded BiasInd"),
```

```
    #"Removed Other Columns" = Table.SelectColumns(#"Removed
Duplicates",{"Analysis group", "Analysis number", "Protocol Outcome", "Scale",
"Subgroup", "Subgroup number", "Applicability", "Study", "Study ID", "Weight",
"Outcome short", "Timepoint (days)", "Short population", "Short comparison", "Overall
risk of bias", "Overall directness"}),
    #"Reordered Columns" = Table.ReorderColumns(#"Removed Other
Columns",{"Analysis group", "Analysis number", "Study", "Protocol Outcome",
"Scale", "Outcome short", "Short population", "Short comparison", "Timepoint (days)",
"Subgroup", "Applicability", "Weight", "Overall risk of bias", "Overall directness"}),
    #"Serious to Moderate" = Table.ReplaceValue(#"Reordered
Columns","Serious","Moderate",Replacer.ReplaceText,{"Overall risk of bias"}),
    #"Very Serious to High" = Table.ReplaceValue(#"Serious to Moderate","Very
Serious","High",Replacer.ReplaceText,{"Overall risk of bias"}),
    #"Changing RoB from text to numbers" = Table.AddColumn(#"Very Serious to
High", "Risk of bias number", each if [Overall risk of bias] = "Low" then 0 else if
[Overall risk of bias] = "Moderate" then 1 else if [Overall risk of bias] = "High" then 2
else "?"),
    #"Removed Other Columns1" = Table.SelectColumns(#"Changing RoB from text
to numbers",{"Analysis group", "Analysis number", "Protocol Outcome", "Scale",
"Subgroup number", "Outcome short", "Study ID", "Weight", "Overall risk of bias",
"Overall directness", "Risk of bias number"}),
    #"Replaced Value" = Table.ReplaceValue(#"Removed Other
Columns1",0,100,Replacer.ReplaceValue,{"Weight"}),
    #"Changing weight null to weight 100" = Table.ReplaceValue(#"Replaced
Value",null,100,Replacer.ReplaceValue,{"Weight"}),
    #"Calculating weighted RoB" = Table.AddColumn(#"Changing weight null to weight
100", "Weighted RoB", each [Risk of bias number] * [Weight], type number),
    #"Dividing by 100" = Table.TransformColumns(#"Calculating weighted RoB",
{{"Weighted RoB", each _ / 100, type number}}),
    #"Removed Other Columns2" = Table.SelectColumns(#"Dividing by
100",{"Analysis group", "Analysis number", "Protocol Outcome", "Scale", "Weighted
RoB"}),
    #"Duplicated Column" = Table.DuplicateColumn(#"Removed Other Columns2",
"Protocol Outcome", "Protocol Outcome - Copy"),
```

#"Pivoted Column to calculate combined RoB" = Table.Pivot(#"Duplicated Column", List.Distinct(#"Duplicated Column"[#"Protocol Outcome - Copy"]), "Protocol Outcome - Copy", "Weighted RoB", List.Sum),
#"Merged Columns1" = Table.CombineColumns(Table.TransformColumnTypes(#"Pivoted Column to calculate combined RoB", {{"All-cause mortality", type text}, {"Thromboembolic events after surgery", type text}, {"Pulmonary embolism", type text}, {"Deep vein thrombosis", type text}, {"Myocardial infarction", type text}, {"Ischaemic stroke", type text}, {"Infection", type text}, {"All-cause readmission", type text}, {"Seizures", type text}, {"Reoperation", type text}}, "en-GB"),{"All-cause mortality", "Thromboembolic events after surgery", "Pulmonary embolism", "Deep vein thrombosis", "Myocardial infarction", "Ischaemic stroke", "Infection", "All-cause readmission", "Seizures", "Reoperation"},Combiner.CombineTextByDelimiter("", QuoteStyle.None),"Summed weighted RoB Number"),
#"Changed Type3" = Table.TransformColumnTypes(#"Merged Columns1",{{"Summed weighted RoB Number", type number}}),
#"Calculation of RoB" = Table.AddColumn(#"Changed Type3", "Outcome RoB", each if [Summed weighted RoB Number] < 0 then "?" else if [Summed weighted RoB Number] <= 0.66 then "Not serious" else if [Summed weighted RoB Number] <= 1.32 then "Serious" else if [Summed weighted RoB Number] <= 2 then "Very serious" else "?"),
#"Analyst manual edit of RoB" = Table.AddColumn(#"Calculation of RoB", "Edited outcome RoB", each if [Protocol Outcome] = "Falls" then "Serious" else null),
#"Risk of Bias final determination" = Table.AddColumn(#"Analyst manual edit of RoB", "RoB", each if [Edited outcome RoB] <> null then [Edited outcome RoB] else [Outcome RoB]),
#"Merged Queries2" = Table.NestedJoin(#"Risk of Bias final determination", {"Analysis group", "Analysis number"}, #"Bias method", {"Analysis group", "Analysis number"}, "Bias method", JoinKind.LeftOuter),
#"Expanded Bias method" = Table.ExpandTableColumn(#"Merged Queries2", "Bias method", {"Method"}, {"Method"}),
#"Risk of bias reason determination" = Table.AddColumn(#"Expanded Bias method", "RoB reason", each if [RoB] = "Serious" then "Risk of bias: Downgraded once. Serious risk of bias in the evidence contributing to the outcomes. More than

```
50% of the weight of the evidence came from studies at moderate or high risk of bias as per" else if [RoB] = "Very serious" then "Risk of bias: Downgraded twice. Very serious risk of bias in the evidence contributing to the outcomes. More than 50% of the weight of the evidence came from studies at high risk of bias as per" else null),
    #"Added Conditional Column" = Table.AddColumn(#"Risk of bias reason determination", "Method merge", each if [RoB reason] = null then null else [Method]),
    #"Merged Columns" = Table.CombineColumns(#"Added Conditional Column",{"RoB reason", "Method merge"},Combiner.CombineTextByDelimiter(" ", QuoteStyle.None),"RoB reason"),
    #"Replaced Value1" = Table.ReplaceValue(#"Merged Columns"," ",null,Replacer.ReplaceValue,{"RoB reason"}),
    #"Inserted Merged Column" = Table.AddColumn(#"Replaced Value1", "GRADE code", each Text.Combine({Text.From([Analysis group], "en-GB"), Text.From([Analysis number], "en-GB")}, "-"), type text),
    #"Removed Other Columns3" = Table.SelectColumns(#"Inserted Merged Column",{"Analysis group", "Analysis number", "RoB", "RoB reason", "GRADE code"}),
    #"Reordered Columns2" = Table.ReorderColumns(#"Removed Other Columns3",{"Analysis group", "Analysis number", "GRADE code", "RoB", "RoB reason"}),
    #"Filtered Rows" = Table.SelectRows(#"Reordered Columns2", each ([Analysis group] <> 5 and [Analysis group] <> 6))
in
    #"Filtered Rows"
```

Bias method

```
let
    Source = Csv.Document(Web.Contents("source"),[Delimiter=","]),
    #"Promoted Headers" = Table.PromoteHeaders(Source, [PromoteAllScalars=true]),
    #"Changed Type" = Table.TransformColumnTypes(#"Promoted Headers",{{"Analysis group", Int64.Type}, {"Analysis number", Int64.Type}, {"Analysis name", type text}, {"Subgroup", type any}, {"Applicability", type text}, {"Study", type text}, {"Study year", Int64.Type}, {"GIV Mean", type number}, {"GIV SE", type
```

number}, {"Experimental mean", type number}, {"Experimental SD", type number}, {"Experimental cases", Int64.Type}, {"Experimental N", Int64.Type}, {"Control mean", type number}, {"Control SD", type number}, {"Control cases", Int64.Type}, {"Control N", Int64.Type}, {"O-E", type any}, {"Variance", type any}, {"Weight", type number}, {"Mean", type number}, {"CI start", type number}, {"CI end", type number}, {"Footnotes", type any}}),

#"Splitting by "" at "" to separate follow up time" = Table.SplitColumn(#"Changed Type", "Analysis name", Splitter.SplitTextByDelimiter(" at ", QuoteStyle.Csv), {"Analysis name.1", "Analysis name.2"}),

#"Changed Type1" = Table.TransformColumnTypes(#"Splitting by "" at "" to separate follow up time",{{"Analysis name.1", type text}, {"Analysis name.2", type text}}),

#"Renamed Columns" = Table.RenameColumns(#"Changed Type1",{{"Analysis name.2", "Follow up time"}}),

#"Splitting by "" ("" to separate scale information" = Table.SplitColumn(#"Renamed Columns", "Analysis name.1", Splitter.SplitTextByDelimiter(" (", QuoteStyle.Csv), {"Analysis name.1.1", "Analysis name.1.2"}),

#"Changed Type2" = Table.TransformColumnTypes(#"Splitting by "" ("" to separate scale information",{{"Analysis name.1.1", type text}, {"Analysis name.1.2", type text}}),

#"Renamed Columns1" = Table.RenameColumns(#"Changed Type2",{{"Analysis name.1.2", "Scale"}, {"Analysis name.1.1", "Protocol Outcome"}}),

#"Removing closing bracket" = Table.ReplaceValue(#"Renamed Columns1",")","",Replacer.ReplaceText,{"Scale"}),

#"Making short outcome" = Table.AddColumn(#"Removing closing bracket", "Outcome short", each if [Protocol Outcome] = "Thromboembolic events after surgery" then "Thromboembolic events" else if [Protocol Outcome] = "All-cause mortality" then "Mortality" else if [Protocol Outcome] = "Pulmonary embolism" then "PE" else if [Protocol Outcome] = "Deep vein thrombosis" then "DVT" else if [Protocol Outcome] = "Myocardial infarction" then "MI" else if [Protocol Outcome] = "Ischaemic stroke" then "Stroke" else if [Protocol Outcome] = "Infection" then "Infection" else if [Protocol Outcome] = "All-cause readmission" then "Readmission" else if [Protocol Outcome] = "Seizures" then "Seizures" else if [Protocol Outcome] = "Reoperation" then "Reoperation" else "?"),

#"Merged Queries" = Table.NestedJoin(#"Making short outcome", {"Study ID"}, #"Study level data", {"Study ID"}, "BiasInd", JoinKind.LeftOuter),
#"Expanded Critical appraisal no edits" = Table.ExpandTableColumn(#"Merged Queries", "BiasInd", {"Follow-up", "Short population", "Short comparison"}, {"Follow-up", "Short population", "Short comparison"}),
#"Renamed Columns2" = Table.RenameColumns(#"Expanded Critical appraisal no edits",{{"Follow-up", "Timepoint (days)"}}),
#"Merged Queries1" = Table.NestedJoin(#"Renamed Columns2", {"Study ID", "Protocol Outcome"}, BiasInd, {"Study ID", "Protocol Outcome"}, "BiasInd", JoinKind.LeftOuter),
#"Expanded BiasInd" = Table.ExpandTableColumn(#"Merged Queries1", "BiasInd", {"Method"}, {"Method"}),
#"Removed Duplicates" = Table.Distinct(#"Expanded BiasInd"),
#"Removed Other Columns" = Table.SelectColumns(#"Removed Duplicates",{"Analysis group", "Analysis number", "Protocol Outcome", "Scale", "Method"}),
#"Reordered Columns" = Table.ReorderColumns(#"Removed Other Columns",{"Analysis group", "Analysis number", "Protocol Outcome", "Scale", "Method"}),
#"Duplicated Column" = Table.DuplicateColumn(#"Reordered Columns", "Protocol Outcome", "Protocol Outcome - Copy"),
#"Method number conversion" = Table.AddColumn(#"Duplicated Column", "Method number", each if [Method] = "ROB 2.0" then 1 else if [Method] = "ROBINS-I" then 2 else if [Method] = "ROBIS" then 3 else "?"),
#"Changed Type4" = Table.TransformColumnTypes(#"Method number conversion",{{"Method number", type number}}),
#"Removed Columns" = Table.RemoveColumns(#"Changed Type4",{"Method"}),
#"Pivoted Column to calculate combined methods" = Table.Pivot(#"Removed Columns", List.Distinct(#"Removed Columns"[#"Protocol Outcome - Copy"]), "Protocol Outcome - Copy", "Method number", List.Average),
#"Merged Columns1" = Table.CombineColumns(Table.TransformColumnTypes(#"Pivoted Column to calculate combined methods", {{"All-cause mortality", type text}, {"Thromboembolic events after surgery", type text}, {"Pulmonary embolism", type text}, {"Deep vein

```
thrombosis", type text}, {"Myocardial infarction", type text}, {"Ischaemic stroke", type text}, {"Infection", type text}, {"All-cause readmission", type text}, {"Seizures", type text}, {"Reoperation", type text}}, "en-GB"),{"All-cause mortality", "Thromboembolic events after surgery", "Pulmonary embolism", "Deep vein thrombosis", "Myocardial infarction", "Ischaemic stroke", "Infection", "All-cause readmission", "Seizures", "Reoperation"},Combiner.CombineTextByDelimiter("", QuoteStyle.None),"Average method number"),
    #"Added Conditional Column" = Table.AddColumn(#"Merged Columns1", "Method", each if [Average method number] = "1" then "ROB 2" else if [Average method number] = "2" then "ROBINS-I" else if [Average method number] = "3" then "ROBIS" else if [Analysis group] <= 3 then "ROB 2 and ROBINS-I" else if [Analysis group] = 4 then "ROB 2 and ROBIS" else "a combination including ROB 2, ROBINS-I and ROBIS")
in
    #"Added Conditional Column"
```

Indirectness

```
let
    Source = Csv.Document(Web.Contents("source"),[Delimiter=","]),
    #"Promoted Headers" = Table.PromoteHeaders(Source, [PromoteAllScalars=true]),
    #"Changed Type" = Table.TransformColumnTypes(#"Promoted Headers",{{"Analysis group", Int64.Type}, {"Analysis number", Int64.Type}, {"Analysis name", type text}, {"Subgroup", type any}, {"Applicability", type text}, {"Study", type text}, {"Study year", Int64.Type}, {"GIV Mean", type number}, {"GIV SE", type number}, {"Experimental mean", type number}, {"Experimental SD", type number}, {"Experimental cases", Int64.Type}, {"Experimental N", Int64.Type}, {"Control mean", type number}, {"Control SD", type number}, {"Control cases", Int64.Type}, {"Control N", Int64.Type}, {"O-E", type any}, {"Variance", type any}, {"Weight", type number}, {"Mean", type number}, {"CI start", type number}, {"CI end", type number}, {"Footnotes", type any}}),
    #"Splitting by "" at "" to separate follow up time" = Table.SplitColumn(#"Changed Type", "Analysis name", Splitter.SplitTextByDelimiter(" at ", QuoteStyle.Csv), {"Analysis name.1", "Analysis name.2"}),
```

```
    #"Changed Type1" = Table.TransformColumnTypes(#"Splitting by "" at "" to separate follow up time",{{"Analysis name.1", type text}, {"Analysis name.2", type text}}),
    #"Renamed Columns" = Table.RenameColumns(#"Changed Type1",{{"Analysis name.2", "Follow up time"}}),
    #"Splitting by "" ("" to separate scale information" = Table.SplitColumn(#"Renamed Columns", "Analysis name.1", Splitter.SplitTextByDelimiter(" (", QuoteStyle.Csv), {"Analysis name.1.1", "Analysis name.1.2"}),
    #"Changed Type2" = Table.TransformColumnTypes(#"Splitting by "" ("" to separate scale information",{{"Analysis name.1.1", type text}, {"Analysis name.1.2", type text}}),
    #"Renamed Columns1" = Table.RenameColumns(#"Changed Type2",{{"Analysis name.1.2", "Scale"}, {"Analysis name.1.1", "Protocol Outcome"}}),
    #"Removing closing bracket" = Table.ReplaceValue(#"Renamed Columns1",")","",Replacer.ReplaceText,{"Scale"}),
    #"Making short outcome" = Table.AddColumn(#"Removing closing bracket", "Outcome short", each if [Protocol Outcome] = "Thromboembolic events after surgery" then "Thromboembolic events" else if [Protocol Outcome] = "All-cause mortality" then "Mortality" else if [Protocol Outcome] = "Pulmonary embolism" then "PE" else if [Protocol Outcome] = "Deep vein thrombosis" then "DVT" else if [Protocol Outcome] = "Myocardial infarction" then "MI" else if [Protocol Outcome] = "Ischaemic stroke" then "Stroke" else if [Protocol Outcome] = "Infection" then "Infection" else if [Protocol Outcome] = "All-cause readmission" then "Readmission" else if [Protocol Outcome] = "Readmission due to thrombosis" then "Thrombosis readmission" else "?"),
    #"Filtered Rows1" = Table.SelectRows(#"Making short outcome", each ([Scale] <> "risk difference")),
    #"Replaced Value2" = Table.ReplaceValue(#"Filtered Rows1","risk ratio",null,Replacer.ReplaceValue,{"Scale"}),
    #"Inserted Merged Column2" = Table.AddColumn(#"Replaced Value2", "Study ID-Outcome-Scale", each Text.Combine({[Study ID], [Protocol Outcome], [Scale]}, "-"), type text),
```

```
    #"Merged Queries" = Table.NestedJoin(#"Inserted Merged Column2", {"Study ID-Outcome-Scale"}, BiasInd, {"Study ID-Outcome-Scale"}, "BiasInd", JoinKind.LeftOuter),
    #"Expanded BiasInd" = Table.ExpandTableColumn(#"Merged Queries", "BiasInd", {"Overall risk of bias", "Overall directness"}, {"Overall risk of bias", "Overall directness"}),
    #"Removed Duplicates" = Table.Distinct(#"Expanded BiasInd"),
    #"Reordered Columns1" = Table.ReorderColumns(#"Removed Duplicates",{"Analysis group", "Analysis number", "Study", "Protocol Outcome", "Scale", "Outcome short", "Subgroup", "Applicability", "Study year", "Weight", "Overall risk of bias", "Overall directness"}),
    #"Removed Other Columns" = Table.SelectColumns(#"Reordered Columns1",{"Analysis group", "Analysis number", "Study", "Protocol Outcome", "Follow up time", "Scale", "Subgroup number", "Outcome short", "Subgroup", "Study ID", "Applicability", "Weight", "Overall risk of bias", "Overall directness"}),
    #"Fully applicable to Directly applicable" = Table.ReplaceValue(#"Removed Other Columns","Fully applicable","Directly applicable",Replacer.ReplaceText,{"Overall directness"}),
    #"Changing directness from text to numbers" = Table.AddColumn(#"Fully applicable to Directly applicable", "Directness number", each if [Overall directness] = "Directly applicable" then 0 else if [Overall directness] = "Partially applicable" then 1 else if [Overall directness] = "Indirectly applicable" then 2 else "?"),
    #"Removed Columns3" = Table.RemoveColumns(#"Changing directness from text to numbers",{"Study", "Subgroup", "Applicability"}),
    #"Replaced Value" = Table.ReplaceValue(#"Removed Columns3",0,100,Replacer.ReplaceValue,{"Weight"}),
    #"Sorted Rows" = Table.Sort(#"Replaced Value",{{"Analysis group", Order.Ascending}, {"Analysis number", Order.Ascending}}),
    #"Calculating weighted directness" = Table.AddColumn(#"Sorted Rows", "Weighted directness", each [Directness number] * [Weight], type number),
    #"Dividing by 100" = Table.TransformColumns(#"Calculating weighted directness", {{"Weighted directness", each _ / 100, type number}}),
```

#"Removed Columns4" = Table.RemoveColumns(#"Dividing by 100",{"Weight", "Overall risk of bias", "Overall directness", "Directness number", "Outcome short", "Study ID", "Subgroup number"}),
#"Duplicated Column" = Table.DuplicateColumn(#"Removed Columns4", "Protocol Outcome", "Protocol Outcome - Copy"),
#"Pivoted Column to calculate combined directness" = Table.Pivot(#"Duplicated Column", List.Distinct(#"Duplicated Column"[#"Protocol Outcome - Copy"]), "Protocol Outcome - Copy", "Weighted directness", List.Sum),
#"Merged Columns" = Table.CombineColumns(Table.TransformColumnTypes(#"Pivoted Column to calculate combined directness", {{"All-cause mortality", type text}, {"Thromboembolic events after surgery", type text}, {"Pulmonary embolism", type text}, {"Deep vein thrombosis", type text}, {"Myocardial infarction", type text}, {"Ischaemic stroke", type text}, {"Infection", type text}, {"All-cause readmission", type text}, {"Seizures", type text}, {"Reoperation", type text}}, "en-GB"),{"All-cause mortality", "Thromboembolic events after surgery", "Pulmonary embolism", "Deep vein thrombosis", "Myocardial infarction", "Ischaemic stroke", "Infection", "All-cause readmission", "Seizures", "Reoperation"},Combiner.CombineTextByDelimiter("", QuoteStyle.None),"Summed weighted directness number"),
#"Changed Type3" = Table.TransformColumnTypes(#"Merged Columns",{{"Summed weighted directness number", type number}}),
#"Calculation of directness" = Table.AddColumn(#"Changed Type3", "Outcome directness", each if [Summed weighted directness number] < 0 then "?" else if [Summed weighted directness number] <= 0.66 then "Not serious" else if [Summed weighted directness number] <= 1.32 then "Serious" else if [Summed weighted directness number] <= 2 then "Very serious" else "?"),
#"Analyst manual edit of directness" = Table.AddColumn(#"Calculation of directness", "Edited outcome directness", each if [Protocol Outcome] = "Falls" then null else null),
#"Directness final determination" = Table.AddColumn(#"Analyst manual edit of directness", "Indirectness", each if [Edited outcome directness] <> null then [Edited outcome directness] else [Outcome directness]),

```
    #"Inserted Merged Column1" = Table.AddColumn(#"Directness final
determination", "Indirectness-outcome", each Text.Combine({[Indirectness], [Protocol
Outcome]}, "-"), type text),
    #"Replaced Value1" = Table.ReplaceValue(#"Inserted Merged Column1","Not
serious","IGNORE",Replacer.ReplaceText,{"Indirectness-outcome"}),
    #"Indirectness reason determination" = Table.AddColumn(#"Replaced Value1",
"Indirectness reason", each if Text.Contains([#"Indirectness-outcome"], "IGNORE")
then null else if [#"Indirectness-outcome"] = "Very serious-Ischaemic stroke" then
"Indirectness: Downgraded twice. Very serious indirectness due to >50% of overall
weighting indirect. Outcome indirectness as people could have had haemorrhagic
strokes as well as ischaemic strokes and at least one study has less than 500
participants in at least one study arm." else if [#"Indirectness-outcome"] = "Serious-
Ischaemic stroke" then "Indirectness: Downgraded once. Serious indirectness due to
>50% of overall weighting partially direct or indirect. Outcome indirectness as people
could have had haemorrhagic strokes as well as ischaemic strokes." else if
[#"Indirectness-outcome"] = "Very serious-Thromboembolic events after surgery"
then "Indirectness: Downgraded twice. Very serious indirectness due to >50% of
overall weighting indirect. Outcome indirectness as thromboembolic events after
surgery aggregate is not consistently including same events  and at least one study
has less than 500 participants in at least one study arm." else if [#"Indirectness-
outcome"] = "Serious-Thromboembolic events after surgery" then "Indirectness:
Downgraded once. Serious indirectness due to >50% of overall weighting partially
direct or indirect. Outcome indirectness as thromboembolic events after surgery
aggregate is not consistently including same events." else if
Text.Contains([#"Indirectness-outcome"], "Serious") then "Indirectness: Downgraded
once. Serious indirectness due to >50% of overall weighting partially direct or
indirect. Outcome indirectness as at least a study contains less than 500 participants
in at least one arm of a study" else null),
    #"Inserted Merged Column" = Table.AddColumn(#"Indirectness reason
determination", "GRADE code", each Text.Combine({Text.From([Analysis group],
"en-GB"), Text.From([Analysis number], "en-GB")}, "-"), type text),
    #"Removed Columns5" = Table.RemoveColumns(#"Inserted Merged
Column",{"Protocol Outcome", "Scale", "Summed weighted directness number",
"Outcome directness", "Edited outcome directness"}),
```

```
    #"Reordered Columns2" = Table.ReorderColumns(#"Removed
Columns5",{"Analysis group", "Analysis number", "GRADE code", "Indirectness",
"Indirectness reason"}),
    #"Filtered Rows" = Table.SelectRows(#"Reordered Columns2", each ([Analysis
group] <> 5 and [Analysis group] <> 6))
in
    #"Filtered Rows"
```

Imprecision

```
let
    Source = Csv.Document(Web.Contents("source"),[Delimiter=","]),
    #"Promoted Headers" = Table.PromoteHeaders(Source,
[PromoteAllScalars=true]),
    #"Changed Type" = Table.TransformColumnTypes(#"Promoted
Headers",{{"Analysis group", Int64.Type}, {"Analysis number", Int64.Type}, {"Analysis
name", type text}, {"Analysis group name", type text}, {"Data source", type text},
{"Data source eligibility", type text}, {"Data type", type text}, {"Log-scale data", type
logical}, {"Outcome", type text}, {"Intervention grouping", type text}, {"Experimental
intervention", type any}, {"Control intervention", type any}, {"Subgroup by", type any},
{"Filter criteria", type any}, {"Experimental group label", type text}, {"Control group
label", type text}, {"Statistical method", type text}, {"Effect measure", type text}, {"Unit
of effect measure", type any}, {"Analysis model", type text}, {"Heterogeneity
estimator", type any}, {"Tau² CI", type logical}, {"Subgroup estimates", type logical},
{"Overall estimates", type logical}, {"Test for subgroup differences", type logical},
{"Prediction interval", type logical}, {"Swap event and non-event", type logical}, {"CI
method", type text}, {"CI/PI level", type number}, {"Experimental cases", Int64.Type},
{"Experimental N", Int64.Type}, {"Control cases", Int64.Type}, {"Control N",
Int64.Type}, {"Mean", type number}, {"CI start", type number}, {"CI end", type
number}, {"PI start", type any}, {"PI end", type any}, {"Heterogeneity Tau²", type any},
{"Tau² CI start", type any}, {"Tau² CI end", type any}, {"Heterogeneity Chi²", type
number}, {"Heterogeneity df", Int64.Type}, {"Heterogeneity P", type number},
{"Heterogeneity I²", type number}, {"Effect Z", type number}, {"Effect P", type
number}, {"Subgroup Chi²", type any}, {"Subgroup df", type any}, {"Subgroup P", type
any}, {"Subgroup I²", type any}, {"ID", type text}}),
```

```
    #"Filtered Rows" = Table.SelectRows(#"Changed Type", each ([Analysis group] <>
5 and [Analysis group] <> 6)),
    #"Removed Other Columns" = Table.SelectColumns(#"Filtered Rows",{"Analysis
group", "Analysis number", "Analysis name", "Analysis group name", "Outcome",
"Experimental intervention", "Control intervention", "Effect measure", "Experimental
cases", "Experimental N", "Control cases", "Control N", "Mean", "CI start", "CI end"}),
    #"Calculating sample size" = Table.AddColumn(#"Removed Other Columns",
"Sample size", each [Experimental N] + [Control N], Int64.Type),
    #"Renamed Columns" = Table.RenameColumns(#"Calculating sample
size",{{"Experimental N", "n1"}, {"Control N", "n2"}, {"Experimental cases", "Number of
events 1"}, {"Control cases", "Number of events 2"}}),
    #"Calculating non-events 1" = Table.AddColumn(#"Renamed Columns", "Non-
events 1", each [n1] - [Number of events 1], Int64.Type),
    #"Calculating non-events 2" = Table.AddColumn(#"Calculating non-events 1",
"Non-events 2", each [n2] - [Number of events 2], Int64.Type),
    #"Calculating Risk1" = Table.AddColumn(#"Calculating non-events 2", "Risk1",
each [Number of events 1] / [n1], type number),
    #"Calculating Risk2" = Table.AddColumn(#"Calculating Risk1", "Risk2", each
[Number of events 2] / [n2], type number),
    #"Calculating risk difference" = Table.AddColumn(#"Calculating Risk2", "Risk
difference", each [Risk1] - [Risk2], type number),
    #"Inserted Multiplication1" = Table.AddColumn(#"Calculating risk difference",
"Event1xNonevent1", each [Number of events 1] * [#"Non-events 1"], Int64.Type),
    #"Inserted Multiplication2" = Table.AddColumn(#"Inserted Multiplication1",
"EventxNonevent2", each [Number of events 2] * [#"Non-events 2"], Int64.Type),
    #"Inserted Cube" = Table.AddColumn(#"Inserted Multiplication2", "n1^3", each
Number.Power([#"Non-events 1"], 3), Int64.Type),
    #"Inserted Cube1" = Table.AddColumn(#"Inserted Cube", "n2^3", each
Number.Power([n2], 3), Int64.Type),
    #"Inserted Division3" = Table.AddColumn(#"Inserted Cube1",
"eventxnonevent1/n1^3", each [Event1xNonevent1] / [#"n1^3"], type number),
    #"Inserted Division4" = Table.AddColumn(#"Inserted Division3", "Division", each
[EventxNonevent2] / [#"n2^3"], type number),
```

#"Renamed Columns1" = Table.RenameColumns(#"Inserted Division4",{{"Division", "eventxnonevent2/n2^3"}}),
#"Inserted Addition2" = Table.AddColumn(#"Renamed Columns1", "eventxnonevent1/n1^3 + eventxnonevent2/n2^3", each [#"eventxnonevent1/n1^3"] + [#"eventxnonevent2/n2^3"], type number),
#"Calculate SE(RD)" = Table.AddColumn(#"Inserted Addition2", "SE(RD)", each Number.Sqrt([#"eventxnonevent1/n1^3 + eventxnonevent2/n2^3"]), type number),
#"Inserted Multiplication3" = Table.AddColumn(#"Calculate SE(RD)", "1.96*SE", each [#"SE(RD)"] * 1.96, type number),
#"Calculate RD 2.5% CI" = Table.AddColumn(#"Inserted Multiplication3", "RD 2.5% CI", each [Risk difference] - [#"1.96*SE"], type number),
#"Calculate RD 97.5% CI" = Table.AddColumn(#"Calculate RD 2.5% CI", "RD 97.5% CI", each [Risk difference] + [#"1.96*SE"], type number),
#"Calculating events/1000 people" = Table.AddColumn(#"Calculate RD 97.5% CI", "Events/1000", each [Risk difference] * 1000, type number),
#"Calculating 2.5% CI" = Table.AddColumn(#"Calculating events/1000 people", "Events/1000 2.5% CI", each [#"RD 2.5% CI"] * 1000, type number),
#"Calculating 97.5% CI" = Table.AddColumn(#"Calculating 2.5% CI", "Events/1000 97.5% CI", each [#"RD 97.5% CI"] * 1000, type number),
#"Changed Type2" = Table.TransformColumnTypes(#"Calculating 97.5% CI",{{"Events/1000 97.5% CI", Int64.Type}, {"Events/1000 2.5% CI", Int64.Type}, {"Events/1000", Int64.Type}}),
#"Removed Other Columns1" = Table.SelectColumns(#"Changed Type2",{"Analysis group", "Analysis number", "Analysis name", "Analysis group name", "Outcome", "Experimental intervention", "Control intervention", "Effect measure", "Number of events 1", "n1", "Number of events 2", "n2", "Mean", "CI start", "CI end", "Sample size", "Events/1000", "Events/1000 2.5% CI", "Events/1000 97.5% CI"}),
#"Subtracting CIs to calculate SE" = Table.AddColumn(#"Removed Other Columns1", "SE", each [CI end] - [CI start], type number),
#"Dividing by 3.92 to calculate SE" = Table.TransformColumns(#"Subtracting CIs to calculate SE", {{"SE", each _ / 3.92, type number}}),
#"Square Root of sample size" = Table.AddColumn(#"Dividing by 3.92 to calculate SE", "Square Root(N)", each Number.Sqrt([Sample size]), type number),

#"Calculate SD" = Table.AddColumn(#"Square Root of sample size", "SD", each [#"Square Root(N)"] * [SE], type number),
#"SD squared" = Table.AddColumn(#"Calculate SD", "SD^2", each Number.Power([SD], 2), type number),
#"Choosing effect for MID precision calc" = Table.AddColumn(#"SD squared", "Events for MIDs", each if [Effect measure] = "Risk Ratio" then [Mean] else if [Effect measure] = "Odds Ratio" then [Mean] else if [Effect measure] = "Risk Difference" then [#"Events/1000"] else [Mean]),
#"Choosing 2.5% CI for MID precision calc" = Table.AddColumn(#"Choosing effect for MID precision calc", "2.5% CI for MIDs", each if [Effect measure] = "Risk Ratio" then [CI start] else if [Effect measure] = "Odds Ratio" then [CI start] else if [Effect measure] = "Risk Difference" then [#"Events/1000 2.5% CI"] else [CI start]),
#"Choosing 97.5% CI for MID precision calc" = Table.AddColumn(#"Choosing 2.5% CI for MID precision calc", "97.5% CI for MIDs", each if [Effect measure] = "Risk Ratio" then [CI end] else if [Effect measure] = "Odds Ratio" then [CI end] else if [Effect measure] = "Risk Difference" then [#"Events/1000 97.5% CI"] else [CI end]),
#"Analyst inputting MIDs" = Table.AddColumn(#"Choosing 97.5% CI for MID precision calc", "MID", each if [Outcome] = "Person/participant generic health-related quality of life - SF-36 physical component score" then 2 else if [Outcome] = "Person/participant generic health-related quality of life - SF-36 mental component score" then 3 else if [Effect measure] = "Risk Ratio" then 0.8 else if [Effect measure] = "Hazard Ratio" then 0.8 else if [Effect measure] = "Odds Ratio" then 0.8 else null),
#"Calculating inverse MIDs (assuming linear scale) - for non-linear scale determine differently" = Table.AddColumn(#"Analyst inputting MIDs", "Inverse MID", each [MID] * -1, type number),
#"Analyst inverse MID choice" = Table.AddColumn(#"Calculating inverse MIDs (assuming linear scale) - for non-linear scale determine differently", "Inverse MID choice", each if [Effect measure] = "Risk Ratio" then 1.25 else if [Effect measure] = "Hazard Ratio" then 1.25 else if [Effect measure] = "Odds Ratio" then 1.25 else [Inverse MID]),
#"2.5% CI relative precision value determination" = Table.AddColumn(#"Analyst inverse MID choice", "2.5% CI MID relation", each if [MID] = null then null else if [#"2.5% CI for MIDs"] >= [Inverse MID choice] then 1 else if [#"2.5% CI for MIDs"] >= [MID] then 0 else if [#"2.5% CI for MIDs"] < [MID] then -1 else null),

#"97.5% CI relative precision value determination" = Table.AddColumn(#"2.5% CI relative precision value determination", "97.5% CI MID relation", each if [MID] = null then null else if [#"97.5% CI for MIDs"] < [Inverse MID choice] then -1 else if [#"97.5% CI for MIDs"] < [MID] then 0 else if [#"97.5% CI for MIDs"] >= [MID] then 1 else null),

#"Changed Type1" = Table.TransformColumnTypes(#"97.5% CI relative precision value determination",{{"2.5% CI MID relation", Int64.Type}, {"97.5% CI MID relation", Int64.Type}}),

#"Combining precision CI values together" = Table.AddColumn(#"Changed Type1", "Precision number", each Text.Combine({Text.From([#"2.5% CI MID relation"], "en-GB"), Text.From([#"97.5% CI MID relation"], "en-GB")}, "="), type text),

#"Replaced Value1" = Table.ReplaceValue(#"Combining precision CI values together","",null,Replacer.ReplaceValue,{"Precision number"}),

#"Imprecision MID calculation" = Table.AddColumn(#"Replaced Value1", "Imprecision MID", each if [Precision number] = null then null else if [Precision number] = "0=0" then "Not serious" else if [Precision number] = "1=1" then "Not serious" else if [Precision number] = "-1=-1" then "Serious" else if [Precision number] = "0=1" then "Serious" else if [Precision number] = "1=0" then "Serious" else if [Precision number] = "-1=0" then "Serious" else if [Precision number] = "0=-1" then "Not serious" else if [Precision number] = "-1=1" then "Very serious" else if [Precision number] = "1=-1" then "Very serious" else "?"),

#"Merged Queries" = Table.NestedJoin(#"Imprecision MID calculation", {"Analysis group", "Analysis number"}, #"OIS dichotomous", {"Analysis group", "Analysis number"}, "OIS dichotomous", JoinKind.LeftOuter),

///If there were continuous outcomes, you would also merge the query for OIS continuous, get the OIS precision for that, merge that with the OIS precision for dichotomous and then do the rest of the steps as usual

#"Expanded OIS dichotomous" = Table.ExpandTableColumn(#"Merged Queries", "OIS dichotomous", {"OIS", "OIS Precision"}, {"OIS", "OIS Precision"}),

#"Replaced Value" = Table.ReplaceValue(#"Expanded OIS dichotomous","Not applicable",null,Replacer.ReplaceValue,{"OIS Precision"}),

#"Analyst imprecision edit" = Table.AddColumn(#"Replaced Value", "Imprecision analyst", each if [Outcome] = "Acute myocardial infarction" then "Very serious" else null),

```
    #"Final imprecision calculation" = Table.AddColumn(#"Analyst imprecision edit", "Imprecision", each if [Imprecision analyst] <> null then [Imprecision analyst] else if [Imprecision MID] <> null then [Imprecision MID] else if [OIS Precision] <> null then [OIS Precision] else null),
    #"Method used to determine precision" = Table.AddColumn(#"Final imprecision calculation", "Method used", each if [Imprecision analyst] <> null then "Imprecision analyst" else if [Imprecision MID] <> null then "Imprecision MID" else if [OIS Precision] <> null then "OIS Precision" else null),
    #"Removing not serious" = Table.AddColumn(#"Method used to determine precision", "Imprecision for reason", each if [Imprecision] = "Not serious" then null else [Imprecision]),
    #"Removing method for not serious" = Table.AddColumn(#"Removing not serious", "Method for reason", each if [Imprecision for reason] = null then null else [Method used]),
    #"Combining method and imprecision" = Table.AddColumn(#"Removing method for not serious", "Method-Imprecision", each Text.Combine({[Method for reason], [Imprecision for reason]}, "-"), type text),
    #"Replaced Value2" = Table.ReplaceValue(#"Combining method and imprecision","",null,Replacer.ReplaceValue,{"Method-Imprecision"}),
    #"Imprecision reason generation" = Table.AddColumn(#"Replaced Value2", "Imprecision reason", each if [Outcome] = "Mortality" then "Downgraded by 1 interval as agreed with committee" else if [#"Method-Imprecision"] = "Imprecision MID-Serious" then "Serious imprecision because 95% CI crosses 1 MID (RR 0.8-1.25)" else if [#"Method-Imprecision"] = "Imprecision MID-Very serious" then "Very serious imprecision because 95% CI crosses 2 MIDs (RR 0.8-1.25)" else if [#"Method-Imprecision"] = "OIS Precision-Serious" then "Serious imprecision because the number of participants is between 30% and 100% of the optimal information size (OIS)" else if [#"Method-Imprecision"] = "OIS Precision-Very serious" then "Very serious imprecision because the number of participants is less than 30% of the optimal information size (OIS) or the ratio between the confidence intervals exceed 2.5 (odds ratio)/3 (risk ratio) or both" else null),
    #"Removed Other Columns2" = Table.SelectColumns(#"Imprecision reason generation",{"Analysis group", "Analysis number", "Analysis name", "Number of events 1", "n1", "Number of events 2", "n2", "Sample size", "Events/1000",
```

```
"Events/1000 2.5% CI", "Events/1000 97.5% CI", "SE", "Square Root(N)", "SD",
"SD^2", "Events for MIDs", "2.5% CI for MIDs", "97.5% CI for MIDs", "Imprecision",
"Imprecision reason"}),
    #"Inserted Merged Column" = Table.AddColumn(#"Removed Other Columns2",
"GRADE group", each Text.Combine({Text.From([Analysis group], "en-GB"),
Text.From([Analysis number], "en-GB")}, "-"), type text)
in
    #"Inserted Merged Column"
```

Number of studies

```
let
    Source = Csv.Document(Web.Contents("source"),[Delimiter=","]),
    #"Promoted Headers" = Table.PromoteHeaders(Source,
[PromoteAllScalars=true]),
    #"Changed Type" = Table.TransformColumnTypes(#"Promoted
Headers",{{"Analysis group", Int64.Type}, {"Analysis number", Int64.Type}, {"Analysis
name", type text}, {"Subgroup", type any}, {"Applicability", type text}, {"Study", type
text}, {"Study year", Int64.Type}, {"GIV Mean", type number}, {"GIV SE", type
number}, {"Experimental mean", type number}, {"Experimental SD", type number},
{"Experimental cases", Int64.Type}, {"Experimental N", Int64.Type}, {"Control mean",
type number}, {"Control SD", type number}, {"Control cases", Int64.Type}, {"Control
N", Int64.Type}, {"O-E", type any}, {"Variance", type any}, {"Weight", type number},
{"Mean", type number}, {"CI start", type number}, {"CI end", type number},
{"Footnotes", type any}}),
    #"Splitting by "" at "" to separate follow up time" = Table.SplitColumn(#"Changed
Type", "Analysis name", Splitter.SplitTextByDelimiter(" at ", QuoteStyle.Csv),
{"Analysis name.1", "Analysis name.2"}),
    #"Changed Type1" = Table.TransformColumnTypes(#"Splitting by "" at "" to
separate follow up time",{{"Analysis name.1", type text}, {"Analysis name.2", type
text}}),
    #"Renamed Columns" = Table.RenameColumns(#"Changed Type1",{{"Analysis
name.2", "Follow up time"}}),
```

```
    #"Splitting by "" ("" to separate scale information" = Table.SplitColumn(#"Renamed
Columns", "Analysis name.1", Splitter.SplitTextByDelimiter(" (", QuoteStyle.Csv),
{"Analysis name.1.1", "Analysis name.1.2"}),
    #"Changed Type2" = Table.TransformColumnTypes(#"Splitting by "" ("" to
separate scale information",{{"Analysis name.1.1", type text}, {"Analysis name.1.2",
type text}}),
    #"Renamed Columns1" = Table.RenameColumns(#"Changed Type2",{{"Analysis
name.1.2", "Scale"}, {"Analysis name.1.1", "Protocol Outcome"}}),
    #"Removing closing bracket" = Table.ReplaceValue(#"Renamed
Columns1",")","",Replacer.ReplaceText,{"Scale"}),
    #"Making short outcome" = Table.AddColumn(#"Removing closing bracket",
"Outcome short", each if [Protocol Outcome] = "Mortality" then "Mort" else if [Protocol
Outcome] = "Physical dependency" then "Pdep" else if [Protocol Outcome] = "Falls"
then "Falls" else if [Protocol Outcome] = "Readmissions to hospital" then "Read" else
if [Protocol Outcome] = "Person/participant generic health-related quality of life" then
"HQOL" else if [Protocol Outcome] = "Activities of daily living" then "ADL" else if
[Protocol Outcome] = "Extended activities of daily living" then "EADL" else if [Protocol
Outcome] = "Length of hospital stay" then "LOS" else if [Protocol Outcome] =
"Caregiver strain index" then "CSI" else if [Protocol Outcome] = "Psychological
distress/mood" then "PD" else if [Protocol Outcome] = "Carer generic health-related
quality of life" then "CHQOL" else if [Protocol Outcome] = "Stroke-specific Patient-
Reported Outcome Measures" then "SQOL" else "?"),
    #"Merged Queries" = Table.NestedJoin(#"Making short outcome", {"Study ID"},
#"Study level data", {"Study ID"}, "Critical appraisal no edits", JoinKind.LeftOuter),
    #"Expanded Critical appraisal no edits1" = Table.ExpandTableColumn(#"Merged
Queries", "Critical appraisal no edits", {"Short population", "Short comparison"},
{"Short population", "Short comparison"}),
    #"Removed Duplicates" = Table.Distinct(#"Expanded Critical appraisal no edits1"),
    #"Removed Other Columns" = Table.SelectColumns(#"Removed
Duplicates",{"Analysis group", "Analysis number", "Protocol Outcome", "Scale",
"Follow up time", "Applicability", "Study"}),
    #"Duplicated Column" = Table.DuplicateColumn(#"Removed Other Columns",
"Protocol Outcome", "Protocol Outcome - Copy"),
```

```
    #"Pivoted Column" = Table.Pivot(#"Duplicated Column", List.Distinct(#"Duplicated Column"[#"Protocol Outcome - Copy"]), "Protocol Outcome - Copy", "Study", List.Count),
    //Redo this each time a new dataset is added in
    #"Inserted Sum" = Table.AddColumn(#"Pivoted Column", "Number of studies", each List.Sum({[#"All-cause mortality"], [Thromboembolic events after surgery], [Pulmonary embolism], [Deep vein thrombosis], [Myocardial infarction], [Ischaemic stroke], [Infection], [#"All-cause readmission"], [Seizures], [Reoperation]})),
    #"Inserted Merged Column" = Table.AddColumn(#"Inserted Sum", "GRADE group", each Text.Combine({Text.From([Analysis group], "en-GB"), Text.From([Analysis number], "en-GB")}, "-"), type text),
    #"Removed Other Columns1" = Table.SelectColumns(#"Inserted Merged Column",{"Analysis group", "Analysis number", "Protocol Outcome", "Scale", "Follow up time", "Applicability", "Number of studies", "GRADE group"}),
    #"Filtered Rows" = Table.SelectRows(#"Removed Other Columns1", each ([Analysis group] <> 5 and [Analysis group] <> 6))
in
    #"Filtered Rows"
```

Inconsistency, units

```
let
    Source = Csv.Document(Web.Contents("source"),[Delimiter=","]),
    #"Promoted Headers" = Table.PromoteHeaders(Source, [PromoteAllScalars=true]),
    #"Changed Type" = Table.TransformColumnTypes(#"Promoted Headers",{{"Analysis group", Int64.Type}, {"Analysis number", Int64.Type}, {"Analysis name", type text}, {"Analysis group name", type text}, {"Data source", type text}, {"Data source eligibility", type text}, {"Data type", type text}, {"Log-scale data", type logical}, {"Outcome", type text}, {"Intervention grouping", type text}, {"Experimental intervention", type any}, {"Control intervention", type any}, {"Subgroup by", type any}, {"Filter criteria", type any}, {"Experimental group label", type text}, {"Control group label", type text}, {"Statistical method", type text}, {"Effect measure", type text}, {"Unit of effect measure", type any}, {"Analysis model", type text}, {"Heterogeneity estimator", type any}, {"Tau² CI", type logical}, {"Subgroup estimates", type logical},
```

{"Overall estimates", type logical}, {"Test for subgroup differences", type logical}, {"Prediction interval", type logical}, {"Swap event and non-event", type logical}, {"CI method", type text}, {"CI/PI level", type number}, {"Experimental cases", Int64.Type}, {"Experimental N", Int64.Type}, {"Control cases", Int64.Type}, {"Control N", Int64.Type}, {"Mean", type number}, {"CI start", type number}, {"CI end", type number}, {"PI start", type any}, {"PI end", type any}, {"Heterogeneity Tau²", type any}, {"Tau² CI start", type any}, {"Tau² CI end", type any}, {"Heterogeneity Chi²", type number}, {"Heterogeneity df", Int64.Type}, {"Heterogeneity P", type number}, {"Heterogeneity I²", type number}, {"Effect Z", type number}, {"Effect P", type number}, {"Subgroup Chi²", type any}, {"Subgroup df", type any}, {"Subgroup P", type any}, {"Subgroup I²", type any}, {"ID", type text}}),
#"Removed Columns" = Table.RemoveColumns(#"Changed Type",{"PI start", "PI end", "Heterogeneity Tau²", "Tau² CI start", "Tau² CI end", "Heterogeneity Chi²", "Heterogeneity df", "Heterogeneity P", "Effect Z", "Effect P", "Subgroup Chi²", "Subgroup df", "Subgroup P", "ID", "CI method", "Test for subgroup differences", "Prediction interval", "Swap event and non-event", "Tau² CI", "Heterogeneity estimator", "Experimental intervention", "Control intervention"}),
#"Replaced Value1" = Table.ReplaceValue(#"Removed Columns",100,0,Replacer.ReplaceValue,{"Heterogeneity I²"}),
#"Inserted Merged Column" = Table.AddColumn(#"Replaced Value1", "GRADE group", each Text.Combine({Text.From([Analysis group], "en-GB"), Text.From([Analysis number], "en-GB")}, "-"), type text),
#"Merged Queries" = Table.NestedJoin(#"Inserted Merged Column", {"GRADE group"}, #"Number of studies", {"GRADE group"}, "Number of studies", JoinKind.LeftOuter),
#"Expanded Number of studies" = Table.ExpandTableColumn(#"Merged Queries", "Number of studies", {"Number of studies"}, {"Number of studies"}),
#"Inconsistency calculation" = Table.AddColumn(#"Expanded Number of studies", "Inconsistency calculation", each if [Number of studies] = 1 then "Serious - single study" else if [#"Heterogeneity I²"] < 0 then "?" else if [#"Heterogeneity I²"] < 40 then "Not serious" else if [#"Heterogeneity I²"] < 60 then "Serious" else if [#"Heterogeneity I²"] < 80 then "Very serious" else if [#"Heterogeneity I²"] < 100 then "Very serious - too heterogenous to meaningfully pool?" else "?"),

#"Analyst edit of inconsistency" = Table.AddColumn(#"Inconsistency calculation", "Inconsistency analyst check", each if [Analysis name] = "Volume of allogenic blood transfused (mL) at end of study" then "Serious" else if [Analysis name] = "Acute myocardial infarction at end of study" then "Serious" else if [Analysis name] = "Length of stay (days) at end of study" then "Very serious" else if [Analysis name] = "Total blood loss volume at end of study" then "Very serious" else null),

#"Final inconsistency calculation" = Table.AddColumn(#"Analyst edit of inconsistency", "Inconsistency", each if [Inconsistency analyst check] <> null then [Inconsistency analyst check] else [Inconsistency calculation]),

#"Inconsistency reason" = Table.AddColumn(#"Final inconsistency calculation", "Inconsistency reason", each if [Number of studies] = 1 then "Inconsistency: Single study- downgraded once for inconsistency, as single study outcomes may otherwise receive favourable ratings for inconsistency by default" else if [#"Heterogeneity I²"] < 40 then null else if [#"Heterogeneity I²"] < 60 then "Inconsistency: Downgraded once. Serious heterogeneity (I2 = 40 to 60%) unexplained by subgroup analysis. Random effects analysis used" else if [#"Heterogeneity I²"] < 80 then "Inconsistency: Downgraded twice. Very serious heterogeneity (serious I2 = >60%) unexplained by subgroup analysis. Random effects analysis used" else if [#"Heterogeneity I²"] <= 100 then "Inconsistency: Downgraded twice. Very serious heterogeneity (serious I2 = >60%) unexplained by subgroup analysis. Random effects analysis used" else if [#"Heterogeneity I²"] > 100 then "?" else if [Analysis name] = "Mortality" then null else null),

#"Removed Columns1" = Table.RemoveColumns(#"Inconsistency reason",{"Inconsistency calculation", "Inconsistency analyst check"}),

#"Analyst editable units column" = Table.AddColumn(#"Removed Columns1", "Units", each if [Unit of effect measure] <> null then [Unit of effect measure] else if [Outcome] = "Length of stay" then "days" else if [Outcome] = "Number of units of allogenic blood transfused" then "units" else if [Outcome] = "Volume of allogenic blood transfused" then "mL" else "Not applicable"),

#"Added Custom1" = Table.AddColumn(#"Analyst editable units column", "|", each "|"),

#"Analyst editable study type" = Table.AddColumn(#"Added Custom1", "Study type", each if Text.Contains([Analysis name], "non-randomised") then "Non-

```
randomised studies" else if Text.Contains([Analysis name], "randomised") then
"Randomised trials" else "Randomised trials"),
    #"Analyst editable ROBINS-I used?" = Table.AddColumn(#"Analyst editable study
type", "ROBINS-I used?", each if [Study type] = "Randomised trials" then false else if
[Outcome] = "a" then true else if [Outcome] = "b" then false else true),
    #"Analyst editable publication bias" = Table.AddColumn(#"Analyst editable
ROBINS-I used?", "Publication bias", each if [Outcome] = "a" then "Undetected" else
if [Outcome] = "b" then "Strongly suspected" else "Undetected"),
    #"Analyst editable large effect" = Table.AddColumn(#"Analyst editable publication
bias", "Large effect", each if [Study type] = "Randomised trials" then "No" else if
[Outcome] = "a" then "No" else if [Outcome] = "b" then "Large" else if [Outcome] = "c"
then "Very large" else "No"),
    #"Analyst editable plausible confounding" = Table.AddColumn(#"Analyst editable
large effect", "Plausible confounding", each if [Study type] = "Randomised trials" then
"No" else if [Outcome] = "a" then "No" else if [Outcome] = "b" then "Would reduce
demonstrated effect" else if [Outcome] = "c" then "Would suggest spurious effect"
else "No"),
    #"Analyst editable dose response gradient" = Table.AddColumn(#"Analyst editable
plausible confounding", "Dose response gradient", each if [Study type] =
"Randomised trials" then "No" else if [Outcome] = "a" then "No" else if [Outcome] =
"b" then "No" else "No"),
    #"Making other considerations column" = Table.AddColumn(#"Analyst editable
dose response gradient", "Other considerations", each Text.Combine({[Publication
bias], [Large effect], [Plausible confounding], [Dose response gradient]}, "
"), type text),
    #"Replaced Value" = Table.ReplaceValue(#"Making other considerations
column","Undetected
No
No
No","None",Replacer.ReplaceText,{"Other considerations"}),
    #"Analyst editable importance" = Table.AddColumn(#"Replaced Value",
"Importance", each if [Outcome] = "Health-related quality of life" then "Critical" else if
[Outcome] = "Mortality" then "Critical" else if [Outcome] = "a" then "Critical" else if
[Outcome] = "b" then "Important" else "Critical"),
```

```
    #"Removed Columns2" = Table.RemoveColumns(#"Analyst editable
importance",{"Publication bias", "Large effect", "Plausible confounding", "Dose
response gradient"}),
    #"Filtered Rows" = Table.SelectRows(#"Removed Columns2", each ([Analysis
group] <> 5 and [Analysis group] <> 6))
in
    #"Filtered Rows"
```

Median baseline rate

```
let
    Source = Csv.Document(Web.Contents("source"),[Delimiter=","]),
    #"Promoted Headers" = Table.PromoteHeaders(Source,
[PromoteAllScalars=true]),
    #"Changed Type" = Table.TransformColumnTypes(#"Promoted
Headers",{{"Analysis group", Int64.Type}, {"Analysis number", Int64.Type}, {"Analysis
name", type text}, {"Experimental cases", Int64.Type}, {"Experimental N",
Int64.Type}, {"Control cases", Int64.Type}, {"Control N", Int64.Type}, {"Mean", type
number}, {"CI start", type number}, {"CI end", type number}}),
    #"Splitting by "" at "" to separate follow up time" = Table.SplitColumn(#"Changed
Type", "Analysis name", Splitter.SplitTextByDelimiter(" at ", QuoteStyle.Csv),
{"Analysis name.1", "Analysis name.2"}),
    #"Changed Type1" = Table.TransformColumnTypes(#"Splitting by "" at "" to
separate follow up time",{{"Analysis name.1", type text}, {"Analysis name.2", type
text}}),
    #"Renamed Columns" = Table.RenameColumns(#"Changed Type1",{{"Analysis
name.2", "Follow up time"}}),
    #"Splitting by "" ("" to separate scale information" = Table.SplitColumn(#"Renamed
Columns", "Analysis name.1", Splitter.SplitTextByDelimiter(" (", QuoteStyle.Csv),
{"Analysis name.1.1", "Analysis name.1.2"}),
    #"Changed Type2" = Table.TransformColumnTypes(#"Splitting by "" ("" to
separate scale information",{{"Analysis name.1.1", type text}, {"Analysis name.1.2",
type text}}),
    #"Renamed Columns1" = Table.RenameColumns(#"Changed Type2",{{"Analysis
name.1.2", "Scale"}, {"Analysis name.1.1", "Protocol Outcome"}}),
```

```
    #"Removing closing bracket" = Table.ReplaceValue(#"Renamed
Columns1",")","",Replacer.ReplaceText,{"Scale"}),
    #"Removed Duplicates" = Table.Distinct(#"Removing closing bracket"),
    #"Inserted Division" = Table.AddColumn(#"Removed Duplicates", "Control group
rate", each [Control cases] / [Control N], type number),
    #"Removed Columns" = Table.RemoveColumns(#"Inserted Division",{"Mean", "CI
start", "CI end", "Experimental cases", "Experimental N", "Control cases", "Control
N"}),
    #"Duplicated Column" = Table.DuplicateColumn(#"Removed Columns", "Protocol
Outcome", "Protocol Outcome - Copy"),
    #"Pivoted Column" = Table.Pivot(#"Duplicated Column", List.Distinct(#"Duplicated
Column"[#"Protocol Outcome - Copy"]), "Protocol Outcome - Copy", "Control group
rate", List.Average),
// Redo each time there is a new dataset
    #"Combining median control events" =
Table.CombineColumns(Table.TransformColumnTypes(#"Pivoted Column", {{"All-
cause mortality", type text}, {"Thromboembolic events after surgery", type text},
{"Pulmonary embolism", type text}, {"Deep vein thrombosis", type text}, {"Myocardial
infarction", type text}, {"Ischaemic stroke", type text}, {"Infection", type text}, {"All-
cause readmission", type text}, {"Seizures", type text}, {"Reoperation", type text}},
"en-GB"),{"All-cause mortality", "Thromboembolic events after surgery", "Pulmonary
embolism", "Deep vein thrombosis", "Myocardial infarction", "Ischaemic stroke",
"Infection", "All-cause readmission", "Seizures",
"Reoperation"},Combiner.CombineTextByDelimiter("", QuoteStyle.None),"Median
control group rate"),
    #"Changed Type3" = Table.TransformColumnTypes(#"Combining median control
events",{{"Median control group rate", type number}}),
    #"Producing GRADE number" = Table.AddColumn(#"Changed Type3", "GRADE
number", each Text.Combine({Text.From([Analysis group], "en-GB"),
Text.From([Analysis number], "en-GB")}, "-"), type text),
    #"Filtered Rows" = Table.SelectRows(#"Producing GRADE number", each
([Analysis group] <> 5 and [Analysis group] <> 6))
in
    #"Filtered Rows"
```

OIS dichotomous

```
let
    Source = Csv.Document(Web.Contents("source"),[Delimiter=","]),
    #"Promoted Headers" = Table.PromoteHeaders(Source,
[PromoteAllScalars=true]),
    #"Changed Type" = Table.TransformColumnTypes(#"Promoted
Headers",{{"Analysis group", Int64.Type}, {"Analysis number", Int64.Type}, {"Analysis
name", type text}, {"Analysis group name", type text}, {"Data source", type text},
{"Data source eligibility", type text}, {"Data type", type text}, {"Log-scale data", type
logical}, {"Outcome", type text}, {"Intervention grouping", type text}, {"Experimental
intervention", type any}, {"Control intervention", type any}, {"Subgroup by", type any},
{"Filter criteria", type any}, {"Experimental group label", type text}, {"Control group
label", type text}, {"Statistical method", type text}, {"Effect measure", type text}, {"Unit
of effect measure", type any}, {"Analysis model", type text}, {"Heterogeneity
estimator", type any}, {"Tau² CI", type logical}, {"Subgroup estimates", type logical},
{"Overall estimates", type logical}, {"Test for subgroup differences", type logical},
{"Prediction interval", type logical}, {"Swap event and non-event", type logical}, {"CI
method", type text}, {"CI/PI level", type number}, {"Experimental cases", Int64.Type},
{"Experimental N", Int64.Type}, {"Control cases", Int64.Type}, {"Control N",
Int64.Type}, {"Mean", type number}, {"CI start", type number}, {"CI end", type
number}, {"PI start", type any}, {"PI end", type any}, {"Heterogeneity Tau²", type any},
{"Tau² CI start", type any}, {"Tau² CI end", type any}, {"Heterogeneity Chi²", type
number}, {"Heterogeneity df", Int64.Type}, {"Heterogeneity P", type number},
{"Heterogeneity I²", type number}, {"Effect Z", type number}, {"Effect P", type
number}, {"Subgroup Chi²", type any}, {"Subgroup df", type any}, {"Subgroup P", type
any}, {"Subgroup I²", type any}, {"ID", type text}}),
    #"Removed Other Columns" = Table.SelectColumns(#"Changed Type",{"Analysis
group", "Analysis number", "Analysis name", "Analysis group name", "Outcome",
"Intervention grouping", "Experimental intervention", "Control intervention",
"Experimental group label", "Control group label", "Effect measure", "Experimental
cases", "Experimental N", "Control cases", "Control N", "Mean", "CI start", "CI end",
"ID"}),
```

```
    #"Removing mean difference and std. mean difference" =
Table.SelectRows(#"Removed Other Columns", each [Effect measure] <> "Mean
Difference" and [Effect measure] <> "Std. Mean Difference"),
    #"Reordered Columns" = Table.ReorderColumns(#"Removing mean difference
and std. mean difference",{"Analysis group", "Analysis number", "ID", "Analysis
name", "Analysis group name", "Outcome", "Intervention grouping", "Experimental
intervention", "Control intervention", "Experimental group label", "Control group label",
"Effect measure", "Experimental cases", "Experimental N", "Control cases", "Control
N"}),
    #"Calculate proportion experimental group" = Table.AddColumn(#"Reordered
Columns", "p1", each [Experimental cases] / [Experimental N], type number),
    #"Calculate proportion control group" = Table.AddColumn(#"Calculate proportion
experimental group", "p2", each [Control cases] / [Control N], type number),
    #"Proportion unexposed with outcomes for GIV" = Table.AddColumn(#"Calculate
proportion control group", "prop unexposed", each 0.5),
    #"Risk ratios only GIV" = Table.AddColumn(#"Proportion unexposed with
outcomes for GIV", "Risk ratio", each if [Effect measure] = "Risk Ratio" then [Mean]
else null),
    #"Odds ratios only GIV" = Table.AddColumn(#"Risk ratios only GIV", "Odds ratio",
each if [Effect measure] = "Odds Ratio" then [Mean] else null),
    #"Risk differences only GIV" = Table.AddColumn(#"Odds ratios only GIV", "Risk
difference", each if [Effect measure] = "Risk difference" then [Mean] else null),
    #"Changed Type2" = Table.TransformColumnTypes(#"Risk differences only
GIV",{{"prop unexposed", type number}, {"Risk ratio", type number}, {"Odds ratio",
type number}, {"Risk difference", type number}}),
    #"OR-1" = Table.AddColumn(#"Changed Type2", "OR-1", each [Odds ratio] - 1,
type number),
    #"proportion unexposed+1" = Table.AddColumn(#"OR-1", "1+pe", each [prop
unexposed] + 1, type number),
    #"Top equation portion" = Table.AddColumn(#"proportion unexposed+1",
"OR*prop unexposed", each [prop unexposed] * [Odds ratio], type number),
    #"Bottom equation portion" = Table.AddColumn(#"Top equation portion", "(OR-
1)*(1+pe)", each [#"1+pe"] * [#"OR-1"], type number),
```

```
    #"Odds ratio proportion exposed GIV" = Table.AddColumn(#"Bottom equation portion", "OR prop exposed", each [#"OR*prop unexposed"] / [#"(OR-1)*(1+pe)"], type number),
    #"Risk ratio proportion exposed GIV" = Table.AddColumn(#"Odds ratio proportion exposed GIV", "RR prop exposed", each [prop unexposed] * [Risk ratio], type number),
    #"Risk difference proportion exposed GIV" = Table.AddColumn(#"Risk ratio proportion exposed GIV", "RD prop exposed", each [Risk difference] + [prop unexposed], type number),
    #"Added Conditional Column" = Table.AddColumn(#"Risk difference proportion exposed GIV", "Proportion exposed", each if [Effect measure] = "Risk Ratio" then [RR prop exposed] else if [Effect measure] = "Odds Ratio" then [OR prop exposed] else if [Effect measure] = "Risk Difference" then [RD prop exposed] else null),
    #"Renamed Columns" = Table.RenameColumns(#"Added Conditional Column",{{"p1", "p1 pre"}, {"p2", "p2 pre"}}),
    #"Added Conditional Column1" = Table.AddColumn(#"Renamed Columns", "p1", each if [p1 pre] = null then [Proportion exposed] else [p1 pre]),
    #"Added Conditional Column2" = Table.AddColumn(#"Added Conditional Column1", "p2", each if [p2 pre] = null then [prop unexposed] else [p2 pre]),
    #"Adding proportions" = Table.AddColumn(#"Added Conditional Column2", "pb", each [p1] + [p2], type number),
    #"Divide added proportions by 2" = Table.TransformColumns(#"Adding proportions", {{"pb", each _ / 2, type number}}),
    #"1" = Table.AddColumn(#"Divide added proportions by 2", "1", each 1),
    #"1-pb" = Table.AddColumn(#"1", "qb", each [1] - [pb], type number),
    #"1-p1" = Table.AddColumn(#"1-pb", "q1", each [1] - [p1], type number),
    #"1-p2" = Table.AddColumn(#"1-p1", "q2", each [1] - [p2], type number),
    #"p1-p2" = Table.AddColumn(#"1-p2", "Delta", each [p1] - [p2], type number),
    #"Absolute only (delta)" = Table.TransformColumns(#"p1-p2",{{"Delta", Number.Abs, type number}}),
    #"za (assuming power 0.8)" = Table.AddColumn(#"Absolute only (delta)", "za", each 0.841621234),
    #"zb (assuming 2 tailed test, alpha 0.05)" = Table.AddColumn(#"za (assuming power 0.8)", "zb", each 1.959963985),
```

```
    #"2" = Table.AddColumn(#"zb (assuming 2 tailed test, alpha 0.05)", "2", each 2),
    #"2pq" = Table.AddColumn(#"2", "sqrt2pq", each List.Product({[pb], [qb], [2]}), type number),
    #"sqrt(2pq)" = Table.TransformColumns(#"2pq",{{"sqrt2pq", Number.Sqrt, type number}}),
    p1q1 = Table.AddColumn(#"sqrt(2pq)", "p1q1", each [p1] * [q1], type number),
    p2q2 = Table.AddColumn(p1q1, "p2q2", each [p2] * [q2], type number),
    sumpqs = Table.AddColumn(p2q2, "sqrtsumpqs", each [p1q1] + [p2q2], type number),
    #"sqrt(sumpqs)" = Table.TransformColumns(sumpqs,{{"sqrtsumpqs", Number.Sqrt, type number}}),
    #"sqrt2pq * za" = Table.AddColumn(#"sqrt(sumpqs)", "sqrt2pq*za", each [sqrt2pq] * [za], type number),
    #"sqrtsumpqs * zb" = Table.AddColumn(#"sqrt2pq * za", "sqrtsumpqs*zb", each [sqrtsumpqs] * [zb], type number),
    #"Adding the two to make the top" = Table.AddColumn(#"sqrtsumpqs * zb", "Top", each [#"sqrt2pq*za"] + [#"sqrtsumpqs*zb"], type number),
    #"Dividing the top by delta" = Table.AddColumn(#"Adding the two to make the top", "sss", each [Top] / [Delta], type number),
    #"Squaring the value" = Table.AddColumn(#"Dividing the top by delta", "OIS", each Number.Power([sss], 2), type number),
    #"Rounding it up to make the OIS" = Table.TransformColumns(#"Squaring the value",{{"OIS", each Number.Round(_, 0), type number}}),
    #"Removed Other Columns1" = Table.SelectColumns(#"Rounding it up to make the OIS",{"Analysis group", "Analysis number", "Analysis name", "Analysis group name", "Outcome", "Intervention grouping", "Experimental intervention", "Control intervention", "Experimental group label", "Control group label", "Effect measure", "Experimental cases", "Experimental N", "Control cases", "CI start", "CI end", "Control N", "OIS"}),
    #"Inserted Addition" = Table.AddColumn(#"Removed Other Columns1", "Total N", each [Experimental N] + [Control N], Int64.Type),
    #"Inserted Multiplication" = Table.AddColumn(#"Inserted Addition", "OIS two arms", each [OIS] * 2, type number),
```

```
    #"OIS proportion calculation" = Table.AddColumn(#"Inserted Multiplication", "Prop OIS", each [Total N] / [OIS two arms], type number),
    #"CI proportion calculation" = Table.AddColumn(#"OIS proportion calculation", "Prop CI", each [CI end] / [CI start], type number),
    #"RR CI proportion conditional" = Table.AddColumn(#"CI proportion calculation", "RR Prop CI", each if [Effect measure] = "Risk Ratio" then [Prop CI] else null),
    #"OR CI proportion conditional" = Table.AddColumn(#"RR CI proportion conditional", "OR Prop CI", each if [Effect measure] = "Odds Ratio" then [Prop CI] else null),
    #"For RR and OR proportions - replace null with 0" = Table.ReplaceValue(#"OR CI proportion conditional",null,0,Replacer.ReplaceValue,{"RR Prop CI", "OR Prop CI"}),
    #"Changed Type1" = Table.TransformColumnTypes(#"For RR and OR proportions - replace null with 0",{{"RR Prop CI", type number}, {"OR Prop CI", type number}}),
    #"Precision conditional determination" = Table.AddColumn(#"Changed Type1", "Precision", each if [Effect measure] = "Hazard Ratio" then "Cannot be determined using this method" else if [RR Prop CI] >= 3 then "Very serious concerns regarding optimal information size" else if [OR Prop CI] >= 2.5 then "Very serious concerns regarding optimal information size" else if [Prop OIS] >= 1 then "No concerns regarding optimal information size" else if [Prop OIS] >= 0.3 then "Serious concerns regarding optimal information size" else if [Prop OIS] < 0.3 then "Very serious concerns regarding optimal information size" else "?"),
    Custom1 = Table.AddColumn(#"Precision conditional determination", "OIS Precision", each if [Effect measure] = "Hazard Ratio" then "Not applicable" else if [RR Prop CI] >= 3 then "Very serious" else if [OR Prop CI] >= 2.5 then "Very serious" else if [Prop OIS] >= 1 then "Not serious" else if [Prop OIS] >= 0.3 then "Serious" else if [Prop OIS] < 0.3 then "Very serious" else "Not serious")
in
    Custom1
```

GRADE table

```
let
    Source = Csv.Document(Web.Contents("source"),[Delimiter=","]),
    #"Promoted Headers" = Table.PromoteHeaders(Source, [PromoteAllScalars=true]),
```

#"Changed Type" = Table.TransformColumnTypes(#"Promoted Headers",{{"Analysis group", Int64.Type}, {"Analysis number", Int64.Type}, {"Analysis name", type text}, {"Subgroup", type any}, {"Applicability", type text}, {"Study", type text}, {"Study year", Int64.Type}, {"GIV Mean", type number}, {"GIV SE", type number}, {"Experimental mean", type number}, {"Experimental SD", type number}, {"Experimental cases", Int64.Type}, {"Experimental N", Int64.Type}, {"Control mean", type number}, {"Control SD", type number}, {"Control cases", Int64.Type}, {"Control N", Int64.Type}, {"O-E", type any}, {"Variance", type any}, {"Weight", type number}, {"Mean", type number}, {"CI start", type number}, {"CI end", type number}, {"Footnotes", type any}}),

#"Splitting by "" at "" to separate follow up time" = Table.SplitColumn(#"Changed Type", "Analysis name", Splitter.SplitTextByDelimiter(" at ", QuoteStyle.Csv), {"Analysis name.1", "Analysis name.2"}),

#"Changed Type1" = Table.TransformColumnTypes(#"Splitting by "" at "" to separate follow up time",{{"Analysis name.1", type text}, {"Analysis name.2", type text}}),

#"Renamed Columns" = Table.RenameColumns(#"Changed Type1",{{"Analysis name.2", "Follow up time"}}),

#"Splitting by "" ("" to separate scale information" = Table.SplitColumn(#"Renamed Columns", "Analysis name.1", Splitter.SplitTextByDelimiter(" (", QuoteStyle.Csv), {"Analysis name.1.1", "Analysis name.1.2"}),

#"Changed Type2" = Table.TransformColumnTypes(#"Splitting by "" ("" to separate scale information",{{"Analysis name.1.1", type text}, {"Analysis name.1.2", type text}}),

#"Renamed Columns1" = Table.RenameColumns(#"Changed Type2",{{"Analysis name.1.2", "Scale"}, {"Analysis name.1.1", "Protocol Outcome"}}),

#"Removing closing bracket" = Table.ReplaceValue(#"Renamed Columns1",")","",Replacer.ReplaceText,{"Scale"}),

#"Making short outcome" = Table.AddColumn(#"Removing closing bracket", "Outcome short", each if [Protocol Outcome] = "Thromboembolic events after surgery" then "Thromboembolic events" else if [Protocol Outcome] = "All-cause mortality" then "Mortality" else if [Protocol Outcome] = "Pulmonary embolism" then "PE" else if [Protocol Outcome] = "Deep vein thrombosis" then "DVT" else if [Protocol Outcome] = "Myocardial infarction" then "MI" else if [Protocol Outcome] = "Ischaemic

```
stroke" then "Stroke" else if [Protocol Outcome] = "Infection" then "Infection" else if [Protocol Outcome] = "All-cause readmission" then "Readmission" else if [Protocol Outcome] = "Readmission due to thrombosis" then "Thrombosis readmission" else "?"),
    #"Merged Queries" = Table.NestedJoin(#"Making short outcome", {"Study ID"}, #"Data rows", {"Study ID"}, "Critical appraisal no edits", JoinKind.LeftOuter),
    #"Expanded Critical appraisal no edits" = Table.ExpandTableColumn(#"Merged Queries", "Critical appraisal no edits", {"Short population", "Short comparison"}, {"Short population", "Short comparison"}),
    #"Removed Duplicates" = Table.Distinct(#"Expanded Critical appraisal no edits"),
    #"Removed Other Columns1" = Table.SelectColumns(#"Removed Duplicates",{"Analysis group", "Analysis number", "Protocol Outcome", "Scale", "Follow up time", "Applicability", "Study", "Study ID"}),
    #"Duplicated Column" = Table.DuplicateColumn(#"Removed Other Columns1", "Protocol Outcome", "Protocol Outcome - Copy"),
    #"Merged Queries5" = Table.NestedJoin(#"Duplicated Column", {"Study ID"}, #"Study level data", {"Study ID"}, "Timepoint", JoinKind.LeftOuter),
    #"Expanded Timepoint" = Table.ExpandTableColumn(#"Merged Queries5", "Timepoint", {"Follow-up"}, {"Follow-up"}),
    #"Renamed Columns3" = Table.RenameColumns(#"Expanded Timepoint",{{"Follow-up", "Average timepoint (days)"}}),
    #"Removed Columns6" = Table.RemoveColumns(#"Renamed Columns3",{"Study ID", "Study"}),
    #"Pivoted Column" = Table.Pivot(#"Removed Columns6", List.Distinct(#"Removed Columns6"[#"Protocol Outcome - Copy"]), "Protocol Outcome - Copy", "Average timepoint (days)", List.Average),
    //Redo this sum with each new dataset
    #"Inserted Sum" = Table.AddColumn(#"Pivoted Column", "Average timepoint (days)", each List.Sum({[#"All-cause mortality"], [Thromboembolic events after surgery], [Pulmonary embolism], [Deep vein thrombosis], [Myocardial infarction], [Ischaemic stroke], [Infection], [#"All-cause readmission"], [Seizures], [Reoperation]}), type number),
    #"Changed Type4" = Table.TransformColumnTypes(#"Inserted Sum",{{"Average timepoint (days)", Int64.Type}}),
```

```
    #"Inserted Merged Column" = Table.AddColumn(#"Changed Type4", "GRADE
group", each Text.Combine({Text.From([Analysis group], "en-GB"),
Text.From([Analysis number], "en-GB")}, "-"), type text),
    #"Removed Other Columns" = Table.SelectColumns(#"Inserted Merged
Column",{"Analysis group", "Analysis number", "Protocol Outcome", "Scale", "Follow
up time", "Applicability", "Average timepoint (days)", "GRADE group"}),
    #"Reordered Columns" = Table.ReorderColumns(#"Removed Other
Columns",{"Analysis group", "Analysis number", "GRADE group", "Protocol
Outcome", "Scale", "Follow up time", "Average timepoint (days)"}),
    #"Filtered Rows" = Table.SelectRows(#"Reordered Columns", each ([Analysis
group] <> 5 and [Analysis group] <> 6)),
    #"Merged Queries1" = Table.NestedJoin(#"Filtered Rows", {"GRADE group"},
#"Risk of bias", {"GRADE code"}, "Risk of bias", JoinKind.LeftOuter),
    #"Expanded Risk of bias" = Table.ExpandTableColumn(#"Merged Queries1", "Risk
of bias", {"RoB", "RoB reason"}, {"RoB", "RoB reason"}),
    #"Merged Queries3" = Table.NestedJoin(#"Expanded Risk of bias", {"GRADE
group"}, #"Inconsistency, units", {"GRADE group"}, "Inconsistency",
JoinKind.LeftOuter),
    #"Expanded Inconsistency" = Table.ExpandTableColumn(#"Merged Queries3",
"Inconsistency", {"Importance", "Inconsistency", "Inconsistency reason", "Other
considerations", "ROBINS-I used?", "Study type", "Units"}, {"Importance",
"Inconsistency.1", "Inconsistency reason", "Other considerations", "ROBINS-I used?",
"Study type", "Units"}),
    #"Renamed Columns2" = Table.RenameColumns(#"Expanded
Inconsistency",{{"Inconsistency.1", "Inconsistency"}}),
    #"Merged Queries2" = Table.NestedJoin(#"Renamed Columns2", {"GRADE
group"}, Indirectness, {"GRADE code"}, "Indirectness", JoinKind.LeftOuter),
    #"Expanded Indirectness" = Table.ExpandTableColumn(#"Merged Queries2",
"Indirectness", {"Indirectness", "Indirectness reason"}, {"Indirectness", "Indirectness
reason"}),
    #"Merged Queries4" = Table.NestedJoin(#"Expanded Indirectness", {"GRADE
group"}, Imprecision, {"GRADE group"}, "Imprecision", JoinKind.LeftOuter),
```

```
    #"Expanded Imprecision" = Table.ExpandTableColumn(#"Merged Queries4",
"Imprecision", {"Imprecision", "Imprecision reason", "Sample size"}, {"Imprecision",
"Imprecision reason", "Sample size"}),
    #"Merged Queries6" = Table.NestedJoin(#"Expanded Imprecision", {"GRADE
group"}, #"Number of studies", {"GRADE group"}, "Number of studies",
JoinKind.LeftOuter),
    #"Expanded Number of studies" = Table.ExpandTableColumn(#"Merged
Queries6", "Number of studies", {"Number of studies"}, {"Number of studies.1"}),
    #"Added Conditional Column1" = Table.AddColumn(#"Expanded Number of
studies", "ROBINS-I adjustment", each if [Study type] = "Randomised trials" then 0
else if [#"ROBINS-I used?"] = true then 0 else 2),
    #"Calculating RoB relative number" = Table.AddColumn(#"Added Conditional
Column1", "RoB number", each if [RoB] = "Not serious" then 0 else if [RoB] =
"Serious" then 1 else if [RoB] = "Very serious" then 2 else "?"),
    #"Calculating indirectness relative number" = Table.AddColumn(#"Calculating RoB
relative number", "Indirectness number", each if [Indirectness] = "Not serious" then 0
else if [Indirectness] = "Serious" then 1 else if [Indirectness] = "Very serious" then 2
else "?"),
    #"Calculating inconsistency number" = Table.AddColumn(#"Calculating
indirectness relative number", "Inconsistency number", each if
Text.StartsWith([Inconsistency], "Not serious") then 0 else if
Text.StartsWith([Inconsistency], "Serious") then 1 else if
Text.StartsWith([Inconsistency], "Very serious") then 2 else "?"),
    #"Calculating imprecision relative number" = Table.AddColumn(#"Calculating
inconsistency number", "Imprecision number", each if [Imprecision] = "Not serious"
then 0 else if [Imprecision] = "Serious" then 1 else if [Imprecision] = "Very serious"
then 2 else "?"),
    #"Changed Type3" = Table.TransformColumnTypes(#"Calculating imprecision
relative number",{{"RoB number", Int64.Type}, {"Indirectness number", Int64.Type},
{"Inconsistency number", Int64.Type}, {"Imprecision number", Int64.Type},
{"ROBINS-I adjustment", Int64.Type}}),
    #"Inserted Sum1" = Table.AddColumn(#"Changed Type3", "GRADE number",
each List.Sum({[#"ROBINS-I adjustment"], [RoB number], [Indirectness number],
[Inconsistency number], [Imprecision number]}), Int64.Type),
```

```
    #"Added Conditional Column" = Table.AddColumn(#"Inserted Sum1", "GRADE",
each if [GRADE number] = 0 then "High" else if [GRADE number] = 1 then
"Moderate" else if [GRADE number] = 2 then "Low" else if [GRADE number] >= 3
then "Very low" else "?"),
    #"Inserted Merged Column1" = Table.AddColumn(#"Added Conditional Column",
"Full reasons", each Text.Combine({[RoB reason], [Inconsistency reason],
[Indirectness reason], [Imprecision reason]}, "
"), type text),
    #"Completing reasons" = Table.AddColumn(#"Inserted Merged Column1",
"Reasons", each if [Full reasons] = "" then "No downgrading required" else [Full
reasons]),
    #"Removed Columns5" = Table.RemoveColumns(#"Completing reasons",{"Full
reasons"}),
    #"Analyst edits for scale column" = Table.AddColumn(#"Removed Columns5",
"Outcome scale", each if [Scale] = "days" then "Not applicable" else if [Scale] = "mL"
then "Not applicable" else if [Scale] = "units" then "Not applicable" else if [Scale] <>
null then [Scale] else if [Protocol Outcome] = "Mortality" then "Not applicable" else
"Not applicable"),
    #"Risk of bias short reason" = Table.AddColumn(#"Analyst edits for scale column",
"Short RoB reason", each if [RoB reason] <> null then "Risk of Bias" else null),
    #"Inconsistency short reason" = Table.AddColumn(#"Risk of bias short reason",
"Inconsistency short reason", each if [Inconsistency reason] <> null then
"Inconsistency" else null),
    #"Indirectness short reason" = Table.AddColumn(#"Inconsistency short reason",
"Indirectness short reason", each if [Indirectness reason] <> null then "Indirectness"
else null),
    #"Imprecision short reason" = Table.AddColumn(#"Indirectness short reason",
"Imprecision short reason", each if [Imprecision reason] <> null then "Imprecision"
else null),
    #"Combining short reasons" = Table.AddColumn(#"Imprecision short reason",
"Short reason", each Text.Combine({[Short RoB reason], [Inconsistency short
reason], [Indirectness short reason], [Imprecision short reason]}, ", "), type text),
```

```
    #"Removed Columns2" = Table.RemoveColumns(#"Combining short
reasons",{"Short RoB reason", "Inconsistency short reason", "Indirectness short
reason", "Imprecision short reason"}),
    #"Added Custom" = Table.AddColumn(#"Removed Columns2", "Scale: ", each
"Scale: "),
    #"Added Custom1" = Table.AddColumn(#"Added Custom", "Units: ", each "Units:
"),
    #"Added Custom2" = Table.AddColumn(#"Added Custom1", "Study types:", each
"Study types: "),
    #"Inserted Merged Column2" = Table.AddColumn(#"Added Custom2", "Scale:
value", each Text.Combine({[#"Scale: "], [Outcome scale]}, ""), type text),
    #"Inserted Merged Column3" = Table.AddColumn(#"Inserted Merged Column2",
"Units: value", each Text.Combine({[#"Units: "], [Units]}, ""), type text),
    #"Inserted Merged Column5" = Table.AddColumn(#"Inserted Merged Column3",
"Study type: value", each Text.Combine({[#"Study types:"], [Study type]}, ""), type
text),
    #"Removed Columns3" = Table.RemoveColumns(#"Inserted Merged
Column5",{"Scale: ", "Units: ", "Study types:"}),
    #"Inserted Merged Column4" = Table.AddColumn(#"Removed Columns3",
"Outcome table", each Text.Combine({[Protocol Outcome], [Follow up time]}, " at "),
type text),
    #"Lowercased Text" = Table.TransformColumns(#"Inserted Merged
Column4",{{"Scale: value", Text.Lower, type text}, {"Units: value", Text.Lower, type
text}, {"Study type: value", Text.Lower, type text}}),
    #"Replaced Value" = Table.ReplaceValue(#"Lowercased
Text","ml","mL",Replacer.ReplaceText,{"Units: value"}),
    #"Merged Columns" = Table.CombineColumns(#"Replaced Value",{"Outcome
table", "Study type: value", "Scale: value", "Units:
value"},Combiner.CombineTextByDelimiter(", ", QuoteStyle.None),"Outcome full
table"),
    #"Added Custom3" = Table.AddColumn(#"Merged Columns", "Risk of bias: ", each
"Risk of bias: "),
    #"Added Custom4" = Table.AddColumn(#"Added Custom3", "Indirectness: ", each
"Indirectness: "),
```

```
    #"Added Custom5" = Table.AddColumn(#"Added Custom4", "Inconsistency: ", each "Inconsistency: "),
    #"Added Custom6" = Table.AddColumn(#"Added Custom5", "Imprecision: ", each "Imprecision: "),
    #"Added Custom7" = Table.AddColumn(#"Added Custom6", "Other concerns: ", each "Other considerations: "),
    #"Inserted Merged Column6" = Table.AddColumn(#"Added Custom7", "Risk of bias: value", each Text.Combine({[#"Risk of bias: "], [RoB]}, ""), type text),
    #"Inserted Merged Column7" = Table.AddColumn(#"Inserted Merged Column6", "Indirectness: value", each Text.Combine({[#"Indirectness: "], [Indirectness]}, ""), type text),
    #"Inserted Merged Column8" = Table.AddColumn(#"Inserted Merged Column7", "Inconsistency: value", each Text.Combine({[#"Inconsistency: "], [Inconsistency]}, ""), type text),
    #"Inserted Merged Column9" = Table.AddColumn(#"Inserted Merged Column8", "Imprecision: value", each Text.Combine({[#"Imprecision: "], [Imprecision]}, ""), type text),
    #"Inserted Merged Column10" = Table.AddColumn(#"Inserted Merged Column9", "Other considerations: value", each Text.Combine({[#"Other concerns: "], [Other considerations]}, ""), type text),
    #"Inserted Merged Column11" = Table.AddColumn(#"Inserted Merged Column10", "GRADE components", each Text.Combine({[#"Risk of bias: value"], [#"Indirectness: value"], [#"Inconsistency: value"], [#"Imprecision: value"], [#"Other considerations: value"]}, "
"), type text),
    #"Removed Columns4" = Table.RemoveColumns(#"Inserted Merged Column11",{"Risk of bias: ", "Indirectness: ", "Inconsistency: ", "Imprecision: ", "Other concerns: ", "Risk of bias: value", "Indirectness: value", "Inconsistency: value", "Imprecision: value", "Other considerations: value"}),
    #"Filtered Rows1" = Table.SelectRows(#"Removed Columns4", each ([Scale] <> "risk difference"))
in
    #"Filtered Rows1"
```