

National Institute for Health and Care Excellence

Blood transfusion (update)

NICE guideline NG24

Methods

November 2025

Draft for Consultation

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 All NICE guidance is subject to regular review and may be updated or withdrawn.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

© NICE 2025. All rights reserved. Subject to [Notice of rights](#).

ISBN: [XXX]

Contents

Development of the guideline	5
What this guideline covers	5
What this guideline does not cover	5
Methods	6
Developing the review questions and outcomes	6
Reviewing research evidence	7
Review protocols	7
Searching for evidence	7
Selecting studies for inclusion	7
Incorporating published evidence syntheses	7
Methods of combining evidence	11
Data synthesis for intervention studies	11
Pairwise meta-analysis	11
Appraising the quality of evidence	14
Intervention studies (relative effect estimates)	14
Minimally important differences (MIDs) and clinical decision thresholds	14
GRADE for intervention studies analysed using pairwise analysis	15
GRADE automation in Power BI	17
Reviewing economic evidence	18
Inclusion and exclusion of economic studies	18
Appraising the quality of economic evidence	18
Health economic modelling	19
References	20
Appendix A Review [A] Power M query for GRADE automation	21
Data rows	21
Overall estimates	24
Risk of bias	34
Indirectness	38
Imprecision	42
Number of studies	48
Timepoint	51
Inconsistency, units	54

Median baseline rate	58
OIS continuous	60
OIS dichotomous	63
GRADE table	68
Appendix B Review [B] Power M query for GRADE automation	75
Data rows	75
Overall estimates	79
Risk of bias	91
Bias method	95
Indirectness	98
Imprecision	103
Number of studies	109
Inconsistency, units	111
Median baseline rate	115
OIS dichotomous	117
GRADE table	121

1 **Development of the guideline**

2 **What this guideline covers**

3 This guideline update discusses:

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

6 **What this guideline does not cover**

7 This guideline did not review evidence on the following:

- 8 • alternatives to blood transfusion (other than tranexamic acid)
- 9 • red blood cells
- 10 • platelets
- 11 • fresh frozen plasma
- 12 • cryoprecipitate
- 13 • prothrombin complex concentrate
- 14 • patient safety
- 15 • patient information
- 16 • blood transfusions for patients with acute upper gastrointestinal bleeding.

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

21
22

1 **Methods**

2 This evidence review was developed using the methods and process described in
3 [Developing NICE guidelines: the manual](#).

4 Declarations of interest were recorded according to [NICE's conflicts of interest policy](#).

5 **Developing the review questions and outcomes**

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

9 The review questions were based on the following frameworks:

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

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

1 **Reviewing research evidence**

2 **Review protocols**

3 Review protocols were developed with the guideline committee to outline the
4 inclusion and exclusion criteria used to select studies for each evidence review.
5 Where possible, review protocols were prospectively registered in the [PROSPERO](#)
6 [register of systematic reviews](#).

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

11 **Searching for evidence**

12 Evidence was searched for each review question using the methods specified in
13 [Developing NICE guidelines: the manual](#).

14 **Selecting studies for inclusion**

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

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

27 **Incorporating published evidence syntheses**

28 If published evidence syntheses were identified sufficiently early in the review
29 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 **Error! Reference source not found..** 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.

1

1 **Methods of combining evidence**

2 **Data synthesis for intervention studies**

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

9 **Pairwise meta-analysis**

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

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

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

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

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

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

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

- 1 were reported using fixed effects models. This may have led to situations where
- 2 pooled results were reported from random-effects models and subgroup results were
- 3 reported from fixed-effects models.
- 4 Where sufficient studies were available, meta-regression was considered to explore
- 5 the effect of study level covariates.

1 **Appraising the quality of evidence**

2 **Intervention studies (relative effect estimates)**

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

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

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

20 **Minimally important differences (MIDs) and clinical decision thresholds**

21 The Core Outcome Measures in Effectiveness Trials (COMET) database was
22 searched to identify published minimal clinically important difference thresholds
23 relevant to this guideline that might aid the committee in identifying clinical decision
24 thresholds for the purpose of GRADE. Identified MIDs were assessed to ensure they
25 had been developed and validated in a methodologically rigorous way, and were
26 applicable to the populations, interventions and outcomes specified in this guideline.
27 In addition, the Guideline Committee were asked to prospectively specify any
28 outcomes where they felt a consensus clinical decision threshold could be defined
29 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.

1 **GRADE automation in Power BI**

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

1 **Reviewing economic evidence**

2 **Inclusion and exclusion of economic studies**

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

11 **Appraising the quality of economic evidence**

12 The cost-utility analysis conducted as part of the work produced by Jaiswal et al.
 13 2025 was appraised using a methodology checklist designed for economic
 14 evaluations (NICE guidelines manual; 2014). This checklist is not intended to judge
 15 the quality of a study per se, but to determine whether an existing economic
 16 evaluation is useful to inform the decision-making of the committee for a specific
 17 topic within the guideline.

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

21 **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

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

4 **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

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

8 **Health economic modelling**

- 9 No original health economic modelling was undertaken for the Blood Transfusion
10 guideline update.
11

1 **References**

- 2 Follmann D, Elliott P, Suh I, Cutler J (1992) Variance imputation for overviews of
3 clinical trials with continuous response. *Journal of Clinical Epidemiology* 45:769–73
- 4 Fu R, Vandermeer BW, Shamliyan TA, et al. (2013) Handling Continuous Outcomes
5 in Quantitative Synthesis In: *Methods Guide for Effectiveness and Comparative*
6 *Effectiveness Reviews* [Internet]. Rockville (MD): Agency for Healthcare Research
7 and Quality (US); 2008-. Available from:
8 <http://www.ncbi.nlm.nih.gov/books/NBK154408/>
- 9 Jaiswal N, Robinson W, Ciminata G et al. (2025) At what levels of expected blood
10 loss from surgery is tranexamic acid (TXA) effective at reducing the need for blood
11 transfusion?
- 12 Norman G., Sloan JA., Wyrwich KW. (2003) Interpretation of changes in health-
13 related quality of life: the remarkable universality of half a standard deviation. *Med*
14 *Care* 41(5):582-92.

1 **Appendix A Review [A] Power M query for GRADE**

2 **automation**

3 **Data rows**

4 let

```
5     Source = Excel.Workbook(Web.Contents("data source"), null, true),
6     #"Data rows_Sheet" = Source[Item="Data rows",Kind="Sheet"][Data],
7     #"Promoted Headers" = Table.PromoteHeaders(#"Data rows_Sheet",
8 [PromoteAllScalars=true]),
9     #"Changed Type" = Table.TransformColumnTypes(#"Promoted
10 Headers",{{"Analysis group", Int64.Type}, {"Analysis number", Int64.Type}, {"Analysis
11 name", type text}, {"Subgroup", type any}, {"Applicability", type text}, {"Study", type
12 text}, {"Study year", Int64.Type}, {"GIV Mean", type number}, {"GIV SE", type
13 number}, {"Experimental mean", type number}, {"Experimental SD", type number},
14 {"Experimental cases", Int64.Type}, {"Experimental N", Int64.Type}, {"Control mean",
15 type number}, {"Control SD", type number}, {"Control cases", Int64.Type}, {"Control
16 N", Int64.Type}, {"O-E", type any}, {"Variance", type any}, {"Weight", type number},
17 {"Mean", type number}, {"CI start", type number}, {"CI end", type number},
18 {"Footnotes", type any}}),
19     #"Splitting by "" at "" to separate follow up time" = Table.SplitColumn(#"Changed
20 Type", "Analysis name", Splitter.SplitTextByDelimiter(" at ", QuoteStyle.Csv),
21 {"Analysis name.1", "Analysis name.2"}),
22     #"Changed Type1" = Table.TransformColumnTypes(#"Splitting by "" at "" to
23 separate follow up time",{{"Analysis name.1", type text}, {"Analysis name.2", type
24 text}}),
25     #"Renamed Columns" = Table.RenameColumns(#"Changed Type1",{{"Analysis
26 name.2", "Follow up time"}}),
27     #"Splitting by "" (" to separate scale information" = Table.SplitColumn(#"Renamed
28 Columns", "Analysis name.1", Splitter.SplitTextByDelimiter(" (", QuoteStyle.Csv),
29 {"Analysis name.1.1", "Analysis name.1.2"}),
30     #"Changed Type2" = Table.TransformColumnTypes(#"Splitting by "" (" to
31 separate scale information",{{"Analysis name.1.1", type text}, {"Analysis name.1.2",
32 type text}}),
```

```

1      #Renamed Columns1" = Table.RenameColumns("#Changed Type2",{{"Analysis
2 name.1.2", "Scale"}, {"Analysis name.1.1", "Protocol Outcome"}}),
3      #Removing closing bracket" = Table.ReplaceValue("#Renamed
4 Columns1",")","",Replacer.ReplaceText,{"Scale"}),
5      #Making short outcome" = Table.AddColumn("#Removing closing bracket",
6 "Outcome short", each if [Protocol Outcome] = "Proportion of patients requiring
7 transfusion" then "Number Transfusions" else if [Protocol Outcome] = "All-cause
8 mortality" then "Mortality" else if [Protocol Outcome] = "Length of stay" then "Length
9 of stay" else if [Protocol Outcome] = "Number of units of allogenic blood transfused"
10 then "Units Transfused" else if [Protocol Outcome] = "Volume of allogenic blood
11 transfused" then "Volume Transfused" else if [Protocol Outcome] = "Serious adverse
12 events" then "Serious AE" else if [Protocol Outcome] = "Thrombotic complications"
13 then "Thrombosis" else if [Protocol Outcome] = "Acute myocardial infarction" then
14 "MI" else if [Protocol Outcome] = "Infection" then "Infection" else "?"),
15      #Merged Queries" = Table.NestedJoin("#Making short outcome", {"Study"},
16 BiasInd, {"Study"}, "BiasInd", JoinKind.LeftOuter),
17      #Expanded Critical appraisal no edits" = Table.ExpandTableColumn("#Merged
18 Queries", "BiasInd", {"Short population", "Short comparison"}, {"Short population",
19 "Short comparison"}),
20      #Removed Duplicates" = Table.Distinct("#Expanded Critical appraisal no edits"),
21      #Removed Columns" = Table.RemoveColumns("#Removed
22 Duplicates,{"Footnotes"}),
23      #Inserted Merged Column" = Table.AddColumn("#Removed Columns", "GRADE
24 number", each Text.Combine({Text.From([Analysis group], "en-GB"),
25 Text.From([Analysis number], "en-GB")}, "-"), type text),
26      #Risk intervention" = Table.AddColumn("#Inserted Merged Column", "Risk
27 Intervention", each [Experimental cases] / [Experimental N], type number),
28      #Risk control" = Table.AddColumn("#Risk intervention", "Risk control", each
29 [Control cases] / [Control N], type number),
30      #Risk ratio" = Table.AddColumn("#Risk control", "Risk ratio", each [Risk
31 Intervention] / [Risk control], type number),
32      #1/a" = Table.AddColumn("#Risk ratio", "1/a", each Number.Power([Experimental
33 cases], -1), Int64.Type),

```

```

1      #"1/c" = Table.AddColumn("#1/a", "1/c", each Number.Power([Control cases], -1),
2      Int64.Type),
3      #"a+b" = Table.AddColumn("#1/c", "a+b", each [Experimental cases] +
4      [Experimental N], Int64.Type),
5      #"c+d" = Table.AddColumn("#a+b", "c+d", each [Control cases] + [Control N],
6      Int64.Type),
7      #"1/a+b" = Table.AddColumn("#c+d", "1/a+b", each Number.Power([#"a+b"], -1),
8      type number),
9      #"1/c+d" = Table.AddColumn("#1/a+b", "1/c+d", each Number.Power([#"c+d"], -1),
10     type number),
11     #"Inserted Natural Logarithm" = Table.AddColumn("#1/c+d", "ln(RR)", each
12     Number.Ln([Risk ratio]), type number),
13     #"Inserted Addition" = Table.AddColumn("#Inserted Natural Logarithm", "1/a +
14     1/c", each [#"1/a"] + [#"1/c"], Int64.Type),
15     #"Inserted Subtraction" = Table.AddColumn("#Inserted Addition", "Subtraction",
16     each [#"1/a + 1/c"] - [#"1/a+b"], type number),
17     #"Inserted Subtraction1" = Table.AddColumn("#Inserted Subtraction",
18     "Subtraction.1", each [Subtraction] - [#"1/c+d"], type number),
19     #"Inserted Square Root" = Table.AddColumn("#Inserted Subtraction1", "Square
20     Root", each Number.Sqrt([Subtraction.1]), type number),
21     #"1.96*SQRT" = Table.AddColumn("#Inserted Square Root", "1.96*SQRT", each
22     [Square Root] * 1.96, type number),
23     #"ln(RR)+1.96*SQRT" = Table.AddColumn("#1.96*SQRT", "Addition", each
24     [#"ln(RR)"] + [#"1.96*SQRT"], type number),
25     #"ln(RR)-1.96*SQRT" = Table.AddColumn("#ln(RR)+1.96*SQRT", "Subtraction.2",
26     each [#"ln(RR)"] - [#"1.96*SQRT"], type number),
27     #"97.5% CI" = Table.AddColumn("#ln(RR)-1.96*SQRT", "97.5% CI", each
28     Number.Exp([Addition]), type number),
29     #"2.5% CI" = Table.AddColumn("#97.5% CI", "2.5% CI", each
30     Number.Exp([Subtraction.2]), type number),
31     #"Removed Columns1" = Table.RemoveColumns("#2.5% CI",{"1/a", "1/c", "a+b",
32     "c+d", "1/a+b", "1/c+d", "ln(RR)", "1/a + 1/c", "Subtraction", "Subtraction.1", "Square
33     Root", "1.96*SQRT", "Addition", "Subtraction.2", "Risk Intervention", "Risk control"}),

```

```

1      #"Added Conditional Column" = Table.AddColumn("#Removed Columns1", "Risk
2      ratio f", each if [Risk ratio] = null then null else if [Risk ratio] >= 0.0001 then [Risk
3      ratio] else null),
4      #"Added Conditional Column1" = Table.AddColumn("#Added Conditional Column",
5      "2.5% CI f", each if [#"2.5% CI"] = null then null else if [#"2.5% CI"] >= 1E-05 then
6      [#"2.5% CI"] else null),
7      #"Added Conditional Column2" = Table.AddColumn("#Added Conditional
8      Column1", "97.5% CI f", each if [#"97.5% CI"] = null then null else if [#"97.5% CI"] >=
9      1E-05 then [#"97.5% CI"] else null),
10     #"Changed Type3" = Table.TransformColumnTypes("#Added Conditional
11     Column2",{{"Risk ratio f", type number}, {"2.5% CI f", type number}, {"97.5% CI f",
12     type number}})
13     in
14     #"Changed Type3"
15

```

16 Overall estimates

```

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

```



```

1  {"Prediction interval", type logical}, {"Swap event and non-event", type logical}, {"CI
2  method", type text}, {"CI/PI level", type number}, {"Experimental cases", Int64.Type},
3  {"Experimental N", Int64.Type}, {"Control cases", Int64.Type}, {"Control N",
4  Int64.Type}, {"Mean", type number}, {"CI start", type number}, {"CI end", type
5  number}, {"PI start", type any}, {"PI end", type any}, {"Heterogeneity Tau2", type any},
6  {"Tau2 CI start", type any}, {"Tau2 CI end", type any}, {"Heterogeneity Chi2", type
7  number}, {"Heterogeneity df", Int64.Type}, {"Heterogeneity P", type number},
8  {"Heterogeneity I2", type number}, {"Effect Z", type number}, {"Effect P", type
9  number}, {"Subgroup Chi2", type any}, {"Subgroup df", type any}, {"Subgroup P", type
10 any}, {"Subgroup I2", type any}, {"ID", type text})),
11   #"Removed Columns" = Table.RemoveColumns(#"Changed Type",{ "PI start", "PI
12   end", "Heterogeneity Tau2", "Tau2 CI start", "Tau2 CI end", "Heterogeneity Chi2",
13   "Heterogeneity df", "Heterogeneity P", "Effect Z", "Effect P", "Subgroup Chi2",
14   "Subgroup df", "Subgroup P", "ID", "CI method", "Test for subgroup differences",
15   "Prediction interval", "Swap event and non-event", "Tau2 CI", "Heterogeneity
16   estimator", "Experimental intervention", "Control intervention"}),
17   #"Inserted Merged Column" = Table.AddColumn(#"Removed Columns", "GRADE
18   group", each Text.Combine({Text.From([Analysis group], "en-GB"),
19   Text.From([Analysis number], "en-GB")}, "-"), type text),
20   #"Analyst editable control risk rate (proportion)" = Table.AddColumn(#"Inserted
21   Merged Column", "Control risk rate (proportion)", each if [Outcome] = "a" then 0.2
22   else null),
23   #"Changed Type6" = Table.TransformColumnTypes(#"Analyst editable control risk
24   rate (proportion)",{{"Control risk rate (proportion)", type number}}),
25   #"Adjusted for editable control n calculation" = Table.AddColumn(#"Changed
26   Type6", "Adjusted control N", each [#"Control risk rate (proportion)"] * [Control N],
27   type number),
28   #"Changed Type8" = Table.TransformColumnTypes(#"Adjusted for editable control
29   n calculation",{{"Adjusted control N", Int64.Type}}),
30   #"Merged Queries" = Table.NestedJoin(#"Changed Type8", {"GRADE group"},
31   #"Median baseline rate", {"GRADE number"}, "Median baseline rate",
32   JoinKind.LeftOuter),

```

```

1      #"Expanded Median baseline rate" = Table.ExpandTableColumn(#"Merged
2      Queries", "Median baseline rate", {"Median control group rate"}, {"Median control
3      group rate"}),
4      #"Adjusted for median control group rate" = Table.AddColumn(#"Expanded
5      Median baseline rate", "Median control n", each [Median control group rate] * [Control
6      N], type number),
7      #"Changed Type7" = Table.TransformColumnTypes(#"Adjusted for median control
8      group rate",{"Median control n", Int64.Type})),
9      #"Renamed Columns4" = Table.RenameColumns(#"Changed Type7",{"Control
10     N", "True Control N"})),
11     #"Analyst editable choice of control group N" = Table.AddColumn(#"Renamed
12     Columns4", "Control N", each if [Outcome] = "a" then [Adjusted control N] else if
13     [Outcome] = "b" then [Median control n] else if [Outcome] = "c" then [True Control N]
14     else [True Control N]),
15     #"Calculating non-events 1" = Table.AddColumn(#"Analyst editable choice of
16     control group N", "Non-events 1", each [Experimental N] - [Experimental cases],
17     Int64.Type),
18     #"Calculating non-events 2" = Table.AddColumn(#"Calculating non-events 1",
19     "Non-events 2", each [Control N] - [Control cases], Int64.Type),
20     #"Calculating control rate" = Table.AddColumn(#"Calculating non-events 2",
21     "Baseline control rate", each ([Control cases] / [Control N]) * 1000, Int64.Type),
22     #"Changed Type9" = Table.TransformColumnTypes(#"Calculating control
23     rate",{"Baseline control rate", Int64.Type})),
24     #"per 1000 people" = Table.AddColumn(#"Changed Type9", "per 1000 people",
25     each " per 1000 people"),
26     #"Making control group rate" =
27     Table.CombineColumns(Table.TransformColumnTypes(#"per 1000 people",
28     {"Baseline control rate", type text}, "en-GB"),{"Baseline control rate", "per 1000
29     people"},Combiner.CombineTextByDelimiter("'", QuoteStyle.None),"Control group
30     rate"),
31     #"Replaced Value" = Table.ReplaceValue(#"Making control group rate"," per 1000
32     people","Not applicable",Replacer.ReplaceValue,{"Control group rate"}),
33     #"Calculating risk1" = Table.AddColumn(#"Replaced Value", "Risk1", each
34     [Experimental cases] / ([Experimental cases] + [Experimental N]), type number),

```

```

1      #"Calculating risk2" = Table.AddColumn(#"Calculating risk1", "Risk2", each
2      [Control cases] / ([Control cases] + [Control N]), type number),
3      #"Renamed Columns" = Table.RenameColumns(#"Calculating
4      risk2",{{"Experimental cases", "Number of events 1"}, {"Experimental N", "n1"},
5      {"Control cases", "Number of events 2"}, {"Control N", "n2"}}),
6      #"Calculating risk difference" = Table.AddColumn(#"Renamed Columns", "Risk
7      difference", each [Risk1] - [Risk2], type number),
8      #"Inserted Multiplication1" = Table.AddColumn(#"Calculating risk difference",
9      "Event1xNonevent1", each [Number of events 1] * [#"Non-events 1"], Int64.Type),
10     #"Inserted Multiplication2" = Table.AddColumn(#"Inserted Multiplication1",
11     "EventxNonevent2", each [Number of events 2] * [#"Non-events 2"], Int64.Type),
12     #"Inserted Cube" = Table.AddColumn(#"Inserted Multiplication2", "n1^3", each
13     Number.Power([#"Non-events 1"], 3), Int64.Type),
14     #"Inserted Cube1" = Table.AddColumn(#"Inserted Cube", "n2^3", each
15     Number.Power([n2], 3), Int64.Type),
16     #"Inserted Division3" = Table.AddColumn(#"Inserted Cube1",
17     "eventxnonevent1/n1^3", each [Event1xNonevent1] / [#"n1^3"], type number),
18     #"Inserted Division4" = Table.AddColumn(#"Inserted Division3", "Division", each
19     [EventxNonevent2] / [#"n2^3"], type number),
20     #"Renamed Columns1" = Table.RenameColumns(#"Inserted
21     Division4",{{"Division", "eventxnonevent2/n2^3"}}),
22     #"Inserted Addition2" = Table.AddColumn(#"Renamed Columns1",
23     "eventxnonevent1/n1^3 + eventxnonevent2/n2^3", each [#"eventxnonevent1/n1^3"] +
24     [#"eventxnonevent2/n2^3"], type number),
25     #"Inserted Square Root1" = Table.AddColumn(#"Inserted Addition2", "SE(RD)",
26     each Number.Sqrt([#"eventxnonevent1/n1^3 + eventxnonevent2/n2^3"]), type
27     number),
28     #"Inserted Multiplication3" = Table.AddColumn(#"Inserted Square Root1",
29     "1.96*SE", each [#"SE(RD)"] * 1.96, type number),
30     #"Inserted Subtraction4" = Table.AddColumn(#"Inserted Multiplication3", "RD 2.5%
31     CI", each [Risk difference] - [#"1.96*SE"], type number),
32     #"Inserted Addition3" = Table.AddColumn(#"Inserted Subtraction4", "RD 97.5%
33     CI", each [Risk difference] + [#"1.96*SE"], type number),

```

```

1      #"Removed Columns3" = Table.RemoveColumns(#"Inserted
2      Addition3",{ "Event1xNonevent1", "EventxNonevent2", "n1^3", "n2^3",
3      "eventxnonevent1/n1^3", "eventxnonevent2/n2^3", "eventxnonevent1/n1^3 +
4      eventxnonevent2/n2^3", "SE(RD)", "1.96*SE", "Non-events 1", "Non-events 2",
5      "Risk1"}),
6      #"Calculating events/1000 people" = Table.AddColumn(#"Removed Columns3",
7      "Events/1000 RD", each [Risk difference] * 1000, type number),
8      #"Calculating 2.5% CI" = Table.AddColumn(#"Calculating events/1000 people",
9      "Events/1000 2.5% CI RD", each [#"RD 2.5% CI"] * 1000, type number),
10     #"Calculating 97.5% CI" = Table.AddColumn(#"Calculating 2.5% CI", "Events/1000
11     97.5% CI RD", each [#"RD 97.5% CI"] * 1000, type number),
12     #"Calculating events/1000 people mean" = Table.AddColumn(#"Calculating 97.5%
13     CI", "Events/1000 mean", each [Mean] * 1000, type number),
14     #"Calculating 2.5% CI mean" = Table.AddColumn(#"Calculating events/1000
15     people mean", "Events/1000 2.5% CI mean", each [#"CI start"] * 1000, type number),
16     #"Calculating 97.5% CI mean" = Table.AddColumn(#"Calculating 2.5% CI mean",
17     "Events/1000 97.5% CI mean", each [#"CI end"] * 1000, type number),
18     #"Changed Type1" = Table.TransformColumnTypes(#"Calculating 97.5% CI
19     mean",{{"Events/1000 97.5% CI RD", Int64.Type}, {"Events/1000 2.5% CI RD",
20     Int64.Type}, {"Events/1000 RD", Int64.Type}, {"Events/1000 97.5% CI mean",
21     Int64.Type}, {"Events/1000 2.5% CI mean", Int64.Type}, {"Events/1000 mean",
22     Int64.Type}}),
23     #"Analyst choice events/1000 method" = Table.AddColumn(#"Changed Type1",
24     "Events/1000", each if [Effect measure] = "Risk Difference" then [#"Events/1000
25     mean"] else if [Analysis name] = "Acute myocardial infarction at end of trial" then
26     [#"Events/1000 mean"] else [#"Events/1000 RD"]),
27     #"Analyst choice 2.5% CI" = Table.AddColumn(#"Analyst choice events/1000
28     method", "Events/1000 2.5% CI", each if [Effect measure] = "Risk Difference" then
29     [#"Events/1000 2.5% CI mean"] else if [Analysis name] = "Acute myocardial infarction
30     at end of study" then [#"Events/1000 2.5% CI mean"] else [#"Events/1000 2.5% CI
31     RD"]),
32     #"Analyst choice 97.5% CI" = Table.AddColumn(#"Analyst choice 2.5% CI",
33     "Events/1000 97.5% CI", each if [Effect measure] = "Risk Difference" then
34     [#"Events/1000 97.5% CI mean"] else if [Analysis name] = "Acute myocardial

```

```

1  infarction at end of study" then [#"Events/1000 97.5% CI mean"] else [#"Events/1000
2  97.5% CI RD"])),
3      #"Extracting RD direction of effect" = Table.AddColumn(#"Analyst choice 97.5%
4  CI", "Sign RD", each Number.Sign([#"Events/1000"]), Int64.Type),
5      #"Extracting 2.5% CI direction of effect" = Table.AddColumn(#"Extracting RD
6  direction of effect", "Sign RD 2.5%", each Number.Sign([#"Events/1000 2.5% CI"]),
7  Int64.Type),
8      #"Extracting 97.5% CI direction of effect" = Table.AddColumn(#"Extracting 2.5% CI
9  direction of effect", "Sign RD 97.5%", each Number.Sign([#"Events/1000 97.5% CI"]),
10 Int64.Type),
11     #"Determining words after RD" = Table.AddColumn(#"Extracting 97.5% CI
12 direction of effect", "RD wording", each if [Sign RD] = 1 then " more events, " else if
13 [Sign RD] = -1 then " fewer events, " else if [Sign RD] = 0 then " fewer events, " else if
14 [Sign RD] = null then null else "?"),
15     #"Determining words after RD 2.5% CI" = Table.AddColumn(#"Determining words
16 after RD", "RD 2.5% CI wording", each if [#"Sign RD 2.5%"] = 1 then " more to " else
17 if [#"Sign RD 2.5%"] = -1 then " fewer to " else if [#"Sign RD 2.5%"] = 0 then " fewer
18 to " else if [#"RD 2.5% CI"] = null then null else "?"),
19     #"Determining words after RD 97.5% CI" = Table.AddColumn(#"Determining
20 words after RD 2.5% CI", "RD 97.5% CI wording", each if [#"Sign RD 97.5%"] = 1
21 then " more" else if [#"Sign RD 97.5%"] = -1 then " fewer" else if [#"Sign RD 97.5%"]
22 = 0 then " fewer" else if [#"Sign RD 97.5%"] = null then null else "?"),
23     #"Maintain RD direction" = Table.DuplicateColumn(#"Determining words after RD
24 97.5% CI", "Events/1000", "Events/1000 direction"),
25     #"Maintain RD 2.5% direction" = Table.DuplicateColumn(#"Maintain RD direction",
26 "Events/1000 2.5% CI", "Events/1000 2.5% CI direction"),
27     #"Maintain RD 97.5% direction" = Table.DuplicateColumn(#"Maintain RD 2.5%
28 direction", "Events/1000 97.5% CI", "Events/1000 97.5% CI direction"),
29     #"Calculated Absolute Value" = Table.TransformColumns(#"Maintain RD 97.5%
30 direction",{{"Events/1000", Number.Abs, Int64.Type}, {"Events/1000 2.5% CI",
31 Number.Abs, Int64.Type}, {"Events/1000 97.5% CI", Number.Abs, Int64.Type}}),
32     #"Combine for written risk difference value" = Table.AddColumn(#"Calculated
33 Absolute Value", "Risk difference effect", each
34 Text.Combine({Text.From([#"Events/1000"], "en-GB"), [RD wording]},

```

```

1 Text.From(["Events/1000 2.5% CI"], "en-GB"), ["RD 2.5% CI wording"],
2 Text.From(["Events/1000 97.5% CI"], "en-GB"), ["RD 97.5% CI wording"], ""), type
3 text),
4     #"Added Conditional Column" = Table.AddColumn(#"Combine for written risk
5 difference value", "Absolute effect", each if [Risk difference effect] <> "" then [Risk
6 difference effect] else if [Effect measure] = "Risk Ratio" then "?" else if [Effect
7 measure] = "Odds Ratio" then "?" else "Not applicable"),
8     #"Multiplied Column" = Table.TransformColumns(#"Added Conditional Column",
9     {"Risk2", each _ * 1000, type number})),
10    #"Renamed Columns2" = Table.RenameColumns(#"Multiplied Column", {"Risk2",
11 "Number of events in control group/1000 people"}),
12    #"Changed Type2" = Table.TransformColumnTypes(#"Renamed
13 Columns2", {"Number of events in control group/1000 people", Int64.Type}),
14    #"Removed Columns2" = Table.RemoveColumns(#"Changed Type2", {"Sign RD",
15 "Sign RD 2.5%", "Sign RD 97.5%"}),
16    #"Inserted Addition" = Table.AddColumn(#"Removed Columns2", "Total events
17 after treatment", each [#"Number of events in control group/1000 people"] +
18 [#"Events/1000 direction"], Int64.Type),
19    #"Determining smallest number of events/1000" = Table.AddColumn(#"Inserted
20 Addition", "Smallest number of events/1000", each if [Total events after treatment] =
21 null then null else if [Total events after treatment] < [#"Number of events in control
22 group/1000 people"] then [Total events after treatment] else [#"Number of events in
23 control group/1000 people"]),
24    #"Changed Type4" = Table.TransformColumnTypes(#"Determining smallest
25 number of events/1000", {"Smallest number of events/1000", Int64.Type}),
26    #"Added Custom" = Table.AddColumn(#"Changed Type4", "1000", each "1000"),
27    #"Inserted Subtraction" = Table.AddColumn(#"Added Custom", "Additional events
28 from treatment", each [Total events after treatment] - [#"Smallest number of
29 events/1000"], type number),
30    #"Inserted Subtraction1" = Table.AddColumn(#"Inserted Subtraction", "Events
31 reduced from treatment", each [#"Number of events in control group/1000 people"] -
32 [#"Smallest number of events/1000"], type number),
33    #"Changed Type3" = Table.TransformColumnTypes(#"Inserted
34 Subtraction1", {"1000", Int64.Type}, {"Smallest number of events/1000", Int64.Type},

```

```

1  {"Additional events from treatment", Int64.Type}, {"Events reduced from treatment",
2  Int64.Type}}),
3    #"Determining largest number of events/1000" = Table.AddColumn(#"Changed
4  Type3", "Largest number of events/1000", each if [#"Number of events in control
5  group/1000 people"] = null then null else if [#"Number of events in control group/1000
6  people"] >= [Total events after treatment] then [#"Number of events in control
7  group/1000 people"] else [Total events after treatment]),
8    #"Changed Type5" = Table.TransformColumnTypes(#"Determining largest number
9  of events/1000",{{"Largest number of events/1000", Int64.Type}}),
10   #"Calculating people without event" = Table.AddColumn(#"Changed Type5",
11   "People without event", each [1000] - [#"Largest number of events/1000"],
12   Int64.Type),
13   #"Renamed Columns3" = Table.RenameColumns(#"Calculating people without
14   event",{{"Smallest number of events/1000", "People with events"}, {"Events reduced
15   from treatment", "Fewer events after treatment"}}),
16   #"Added Custom1" = Table.AddColumn(#"Renamed Columns3", "|", each "|"),
17   #"Filtered Rows" = Table.SelectRows(#"Added Custom1", each ([Analysis group] =
18   1)),
19   #"Added Custom2" = Table.AddColumn(#"Filtered Rows", "(", each "("),
20   #"Added Custom3" = Table.AddColumn(#"Added Custom2", ")", each ")",
21   #"Rounded Off" = Table.TransformColumns(#"Added Custom3",{{"Mean", each
22   Number.Round(_, 5), type number}, {"CI start", each Number.Round(_, 5), type
23   number}, {"CI end", each Number.Round(_, 5), type number}}),
24   #"Inserted Merged Column1" = Table.AddColumn(#"Rounded Off", "Effect size",
25   each Text.Combine({Text.From([Mean], "en-GB"), [#("("], " " ), type text),
26   #"Inserted Merged Column2" = Table.AddColumn(#"Inserted Merged Column1",
27   "Effect Size.1", each Text.Combine({[Effect size], Text.From([CI start], "en-GB")}, ""),
28   type text),
29   #"Inserted Merged Column3" = Table.AddColumn(#"Inserted Merged Column2",
30   "Effect size.2", each Text.Combine({[Effect Size.1], Text.From([CI end], "en-GB")}, " ",
31   type text),
32   #"Inserted Merged Column4" = Table.AddColumn(#"Inserted Merged Column3",
33   "Effect size.3", each Text.Combine({[Effect size.2], [#(")"]}, ""), type text),

```

```

1      #"Removed Columns1" = Table.RemoveColumns(#"Inserted Merged
2      Column4",{("(", ")","Effect size", "Effect Size.1", "Effect size.2"}),
3      #"Renamed Columns5" = Table.RenameColumns(#"Removed Columns1",{{"Effect
4      size.3", "Effect size"}}),
5      #"Risk intervention" = Table.AddColumn(#"Renamed Columns5", "Risk
6      Intervention", each [Number of events 1] / [n1], type number),
7      #"Risk control" = Table.AddColumn(#"Risk intervention", "Risk control", each
8      [Number of events 2] / [True Control N], type number),
9      #"Risk ratio" = Table.AddColumn(#"Risk control", "Risk ratio", each [Risk
10     Intervention] / [Risk control], type number),
11     #"1/a" = Table.AddColumn(#"Risk ratio", "1/a", each Number.Power([Number of
12     events 1], -1), Int64.Type),
13     #"1/c" = Table.AddColumn(#"1/a", "1/c", each Number.Power([Number of events
14     2], -1), Int64.Type),
15     #"a+b" = Table.AddColumn(#"1/c", "a+b", each [Number of events 1] + [n1],
16     Int64.Type),
17     #"c+d" = Table.AddColumn(#"a+b", "c+d", each [Number of events 2] + [True
18     Control N], Int64.Type),
19     #"1/a+b" = Table.AddColumn(#"c+d", "1/a+b", each Number.Power([#"a+b"], -1),
20     type number),
21     #"1/c+d" = Table.AddColumn(#"1/a+b", "1/c+d", each Number.Power([#"c+d"], -1),
22     type number),
23     #"Inserted Natural Logarithm" = Table.AddColumn(#"1/c+d", "ln(RR)", each
24     Number.Ln([Risk ratio]), type number),
25     #"Inserted Addition4" = Table.AddColumn(#"Inserted Natural Logarithm", "1/a +
26     1/c", each [#"1/a"] + [#"1/c"], Int64.Type),
27     #"Inserted Subtraction2" = Table.AddColumn(#"Inserted Addition4", "Subtraction",
28     each [#"1/a + 1/c"] - [#"1/a+b"], type number),
29     #"Inserted Subtraction3" = Table.AddColumn(#"Inserted Subtraction2",
30     "Subtraction.1", each [Subtraction] - [#"1/c+d"], type number),
31     #"Inserted Square Root" = Table.AddColumn(#"Inserted Subtraction3", "Square
32     Root", each Number.Sqrt([Subtraction.1]), type number),
33     #"1.96*SQRT" = Table.AddColumn(#"Inserted Square Root", "1.96*SQRT", each
34     [Square Root] * 1.96, type number),

```



```

1      #"ln(RR)+1.96*SQRT" = Table.AddColumn("#1.96*SQRT", "Addition", each
2      [#"ln(RR)"] + [#"1.96*SQRT"], type number),
3      #"ln(RR)-1.96*SQRT" = Table.AddColumn("#ln(RR)+1.96*SQRT", "Subtraction.2",
4      each [#"ln(RR)"] - [#"1.96*SQRT"], type number),
5      #"97.5% CI" = Table.AddColumn("#ln(RR)-1.96*SQRT", "97.5% CI", each
6      Number.Exp([Addition]), type number),
7      #"2.5% CI" = Table.AddColumn("#97.5% CI", "2.5% CI", each
8      Number.Exp([Subtraction.2]), type number),
9      #"Removed Columns4" = Table.RemoveColumns("#2.5% CI",{"1/a", "1/c", "a+b",
10     "c+d", "1/a+b", "1/c+d", "ln(RR)", "1/a + 1/c", "Subtraction", "Subtraction.1", "Square
11     Root", "1.96*SQRT", "Addition", "Subtraction.2", "Risk Intervention", "Risk control"}),
12     #"Added Conditional Column1" = Table.AddColumn("#Removed Columns4", "Risk
13     ratio f", each if [Risk ratio] = null then null else if [Risk ratio] >= 0.0001 then [Risk
14     ratio] else null),
15     #"Added Conditional Column2" = Table.AddColumn("#Added Conditional
16     Column1", "2.5% CI f", each if [#"2.5% CI"] = null then null else if [#"2.5% CI"] >= 1E-
17     05 then [#"2.5% CI"] else null),
18     #"Added Conditional Column3" = Table.AddColumn("#Added Conditional
19     Column2", "97.5% CI f", each if [#"97.5% CI"] = null then null else if [#"97.5% CI"] >=
20     1E-05 then [#"97.5% CI"] else null),
21     #"Changed Type0" = Table.TransformColumnTypes("#Added Conditional
22     Column3",{"Risk ratio f", type number}, {"2.5% CI f", type number}, {"97.5% CI f",
23     type number})),
24     #"Added Conditional Column4" = Table.AddColumn("#Changed Type0", "Risk
25     differencef", each if [Effect measure] = "Risk Difference" then [Mean] else [Risk
26     difference]),
27     #"Added Conditional Column5" = Table.AddColumn("#Added Conditional
28     Column4", "RD 2.5% CI f", each if [Effect measure] = "Risk Difference" then [CI start]
29     else [#"RD 2.5% CI"]),
30     #"Added Conditional Column6" = Table.AddColumn("#Added Conditional
31     Column5", "RD 97.5% CI f", each if [Effect measure] = "Risk Difference" then [CI end]
32     else [#"RD 97.5% CI"]),

```

```

1      #"Changed Type10" = Table.TransformColumnTypes(#"Added Conditional
2      Column6",{{"Risk difference", type number}, {"RD 2.5% CI f", type number}, {"RD
3      97.5% CI f", type number}})

```

```

4      in

```

```

5      #"Changed Type10"
6

```

7 Risk of bias

```

8      let

```

```

9          Source = Excel.Workbook(Web.Contents("source"), null, true),

```

```

10         #"Data rows_Sheet" = Source[[Item="Data rows",Kind="Sheet"]][Data],

```

```

11         #"Promoted Headers" = Table.PromoteHeaders(#"Data rows_Sheet",

```

```

12         [PromoteAllScalars=true]),

```

```

13         #"Changed Type" = Table.TransformColumnTypes(#"Promoted

```

```

14         Headers",{{"Analysis group", Int64.Type}, {"Analysis number", Int64.Type}, {"Analysis

```

```

15         name", type text}, {"Subgroup", type any}, {"Applicability", type text}, {"Study", type

```

```

16         text}, {"Study year", Int64.Type}, {"GIV Mean", type number}, {"GIV SE", type

```

```

17         number}, {"Experimental mean", type number}, {"Experimental SD", type number},

```

```

18         {"Experimental cases", Int64.Type}, {"Experimental N", Int64.Type}, {"Control mean",

```

```

19         type number}, {"Control SD", type number}, {"Control cases", Int64.Type}, {"Control

```

```

20         N", Int64.Type}, {"O-E", type any}, {"Variance", type any}, {"Weight", type number},

```

```

21         {"Mean", type number}, {"CI start", type number}, {"CI end", type number},

```

```

22         {"Footnotes", type any}}),

```

```

23         #"Splitting by "" at "" to separate follow up time" = Table.SplitColumn(#"Changed

```

```

24         Type", "Analysis name", Splitter.SplitTextByDelimiter(" at ", QuoteStyle.Csv),

```

```

25         {"Analysis name.1", "Analysis name.2"}),

```

```

26         #"Changed Type1" = Table.TransformColumnTypes(#"Splitting by "" at "" to

```

```

27         separate follow up time",{{"Analysis name.1", type text}, {"Analysis name.2", type

```

```

28         text}}),

```

```

29         #"Renamed Columns" = Table.RenameColumns(#"Changed Type1",{{"Analysis

```

```

30         name.2", "Follow up time"}),

```

```

31         #"Splitting by "" (" to separate scale information" = Table.SplitColumn(#"Renamed

```

```

32         Columns", "Analysis name.1", Splitter.SplitTextByDelimiter(" (", QuoteStyle.Csv),

```

```

33         {"Analysis name.1.1", "Analysis name.1.2"}),

```

```

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

```

```

1 population", "Short comparison", "Timepoint (months)", "Follow up time", "Subgroup",
2 "Applicability", "Study", "Study year", "Weight", "Overall risk of bias", "Overall
3 directness"}),
4   #"Removed Columns1" = Table.RemoveColumns(#"Reordered Columns",{ "Follow
5 up time"}),
6   #"Reordered Columns1" = Table.ReorderColumns(#"Removed
7 Columns1",{ "Analysis group", "Analysis number", "Study", "Protocol Outcome",
8 "Scale", "Outcome short", "Short population", "Short comparison", "Timepoint
9 (months)", "Subgroup", "Applicability", "Study year", "Weight", "Overall risk of bias",
10 "Overall directness"}),
11   #"Removed Columns2" = Table.RemoveColumns(#"Reordered Columns1",{ "Study
12 year"}),
13   #"Changing RoB from text to numbers" = Table.AddColumn(#"Removed
14 Columns2", "Risk of bias number", each if [Overall risk of bias] = "Low" then 0 else if
15 [Overall risk of bias] = "Some concerns" then 1 else if [Overall risk of bias] = "High"
16 then 2 else "?"),
17   #"Removed Columns3" = Table.RemoveColumns(#"Changing RoB from text to
18 numbers",{ "Study", "Short population", "Short comparison", "Subgroup",
19 "Applicability", "Timepoint (months)"}),
20   #"Calculating weighted RoB" = Table.AddColumn(#"Removed Columns3",
21 "Weighted RoB", each [Risk of bias number] * [Weight], type number),
22   #"Dividing by 100" = Table.TransformColumns(#"Calculating weighted RoB",
23 {{ "Weighted RoB", each _ / 100, type number }}),
24   #"Removed Columns4" = Table.RemoveColumns(#"Dividing by 100",{ "Weight",
25 "Overall risk of bias", "Overall directness", "Risk of bias number"}),
26   #"Duplicated Column" = Table.DuplicateColumn(#"Removed Columns4", "Protocol
27 Outcome", "Protocol Outcome - Copy"),
28   #"Pivoted Column to calculate combined RoB" = Table.Pivot(#"Duplicated
29 Column", List.Distinct(#"Duplicated Column"[#"Protocol Outcome - Copy"]), "Protocol
30 Outcome - Copy", "Weighted RoB", List.Sum),
31   #"Merged Columns" =
32 Table.CombineColumns(Table.TransformColumnTypes(#"Pivoted Column to
33 calculate combined RoB", {{ "Proportion of patients requiring transfusion", type text},
34 {"Thrombotic complications", type text}, {"Duplicate outcome proportion of people

```

```

1 requiring transfusion", type text}, {"Serious adverse events", type text}, {"All-cause
2 mortality", type text}, {"Number of units of allogenic blood transfused", type text},
3 {"Infection", type text}, {"Length of stay", type text}, {"Acute myocardial infarction",
4 type text}, {"Volume of allogenic blood transfused", type text}, {"Total blood loss
5 volume", type text}}, "en-GB"), {"Proportion of patients requiring transfusion",
6 "Thrombotic complications", "Duplicate outcome proportion of people requiring
7 transfusion", "Serious adverse events", "All-cause mortality", "Number of units of
8 allogenic blood transfused", "Infection", "Length of stay", "Acute myocardial
9 infarction", "Volume of allogenic blood transfused", "Total blood loss
10 volume"}, Combiner.CombineTextByDelimiter("", QuoteStyle.None), "Summed
11 weighted RoB Number"),
12   #"Changed Type3" = Table.TransformColumnTypes(#"Merged
13 Columns", {"Summed weighted RoB Number", type number})),
14   #"Calculation of RoB" = Table.AddColumn(#"Changed Type3", "Outcome RoB",
15 each if [Summed weighted RoB Number] < 0 then "?" else if [Summed weighted RoB
16 Number] <= 0.66 then "Not serious" else if [Summed weighted RoB Number] <= 1.32
17 then "Serious" else if [Summed weighted RoB Number] <= 2 then "Very serious" else
18 "?"),
19   #"Analyst manual edit of RoB" = Table.AddColumn(#"Calculation of RoB", "Edited
20 outcome RoB", each if [Protocol Outcome] = "Falls" then "Serious" else null),
21   #"Risk of Bias final determination" = Table.AddColumn(#"Analyst manual edit of
22 RoB", "RoB", each if [Edited outcome RoB] <> null then [Edited outcome RoB] else
23 [Outcome RoB]),
24   #"Risk of bias reason determination" = Table.AddColumn(#"Risk of Bias final
25 determination", "RoB reason", each if [RoB] = "Serious" then "Risk of bias:
26 Downgraded once. Serious risk of bias in the evidence contributing to the outcomes.
27 More than 50% of the weight of the evidence came from studies at moderate or high
28 risk of bias as per RoB2" else if [RoB] = "Very serious" then "Risk of bias:
29 Downgraded twice. Very serious risk of bias in the evidence contributing to the
30 outcomes. More than 50% of the weight of the evidence came from studies at high
31 risk of bias as per RoB2" else null),
32   #"Inserted Merged Column" = Table.AddColumn(#"Risk of bias reason
33 determination", "GRADE code", each Text.Combine({Text.From([Analysis group],
34 "en-GB"), Text.From([Analysis number], "en-GB")}, "-"), type text),

```

```

1      #"Removed Columns5" = Table.RemoveColumns(#"Inserted Merged
2      Column",{ "Protocol Outcome", "Scale", "Outcome short", "Summed weighted RoB
3      Number", "Outcome RoB", "Edited outcome RoB"}),
4      #"Reordered Columns2" = Table.ReorderColumns(#"Removed
5      Columns5",{ "Analysis group", "Analysis number", "GRADE code", "RoB", "RoB
6      reason"})
7      in
8      #"Reordered Columns2"
9

```

10 Indirectness

```

11      let
12          Source = Excel.Workbook(Web.Contents("source"), null, true),
13          #"Data rows_Sheet" = Source[[Item="Data rows",Kind="Sheet"]][Data],
14          #"Promoted Headers" = Table.PromoteHeaders(#"Data rows_Sheet",
15      [PromoteAllScalars=true]),
16          #"Changed Type" = Table.TransformColumnTypes(#"Promoted
17      Headers",{{"Analysis group", Int64.Type}, {"Analysis number", Int64.Type}, {"Analysis
18      name", type text}, {"Subgroup", type any}, {"Applicability", type text}, {"Study", type
19      text}, {"Study year", Int64.Type}, {"GIV Mean", type number}, {"GIV SE", type
20      number}, {"Experimental mean", type number}, {"Experimental SD", type number},
21      {"Experimental cases", Int64.Type}, {"Experimental N", Int64.Type}, {"Control mean",
22      type number}, {"Control SD", type number}, {"Control cases", Int64.Type}, {"Control
23      N", Int64.Type}, {"O-E", type any}, {"Variance", type any}, {"Weight", type number},
24      {"Mean", type number}, {"CI start", type number}, {"CI end", type number},
25      {"Footnotes", type any}}),
26          #"Splitting by "" at "" to separate follow up time" = Table.SplitColumn(#"Changed
27      Type", "Analysis name", Splitter.SplitTextByDelimiter(" at ", QuoteStyle.Csv),
28      {"Analysis name.1", "Analysis name.2"}),
29          #"Changed Type1" = Table.TransformColumnTypes(#"Splitting by "" at "" to
30      separate follow up time",{{"Analysis name.1", type text}, {"Analysis name.2", type
31      text}}),
32          #"Renamed Columns" = Table.RenameColumns(#"Changed Type1",{{"Analysis
33      name.2", "Follow up time"}}),

```

```

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

```

```

1      #"Timepoint = 0" = Table.AddColumn(#"Removed Columns", "Timepoint
2      (months)", each 0),
3      #"Removed Columns1" = Table.RemoveColumns(#"Timepoint = 0",{"Follow up
4      time"}),
5      #"Reordered Columns1" = Table.ReorderColumns(#"Removed
6      Columns1",{"Analysis group", "Analysis number", "Study", "Protocol Outcome",
7      "Scale", "Outcome short", "Short population", "Short comparison", "Timepoint
8      (months)", "Subgroup", "Applicability", "Study year", "Weight", "Overall risk of bias",
9      "Overall directness"}),
10     #"Removed Columns2" = Table.RemoveColumns(#"Reordered Columns1",{"Study
11     year"}),
12     #"Changing directness from text to numbers" = Table.AddColumn(#"Removed
13     Columns2", "Directness number", each if [Overall directness] = "Directly applicable"
14     then 0 else if [Overall directness] = "Partially applicable" then 1 else if [Overall
15     directness] = "Indirectly applicable" then 2 else "?"),
16     #"Removed Columns3" = Table.RemoveColumns(#"Changing directness from text
17     to numbers",{"Study", "Short population", "Short comparison", "Timepoint (months)",
18     "Subgroup", "Applicability"}),
19     #"Calculating weighted directness" = Table.AddColumn(#"Removed Columns3",
20     "Weighted directness", each [Directness number] * [Weight], type number),
21     #"Dividing by 100" = Table.TransformColumns(#"Calculating weighted directness",
22     {"Weighted directness", each _ / 100, type number}),
23     #"Removed Columns4" = Table.RemoveColumns(#"Dividing by 100",{"Weight",
24     "Overall risk of bias", "Overall directness", "Directness number"}),
25     #"Duplicated Column" = Table.DuplicateColumn(#"Removed Columns4", "Protocol
26     Outcome", "Protocol Outcome - Copy"),
27     #"Pivoted Column to calculate combined directness" = Table.Pivot(#"Duplicated
28     Column", List.Distinct(#"Duplicated Column"[#"Protocol Outcome - Copy"]), "Protocol
29     Outcome - Copy", "Weighted directness", List.Sum),
30     #"Merged Columns" =
31     Table.CombineColumns(Table.TransformColumnTypes(#"Pivoted Column to
32     calculate combined directness", {"Proportion of patients requiring transfusion", type
33     text}, {"Thrombotic complications", type text}, {"Duplicate outcome proportion of
34     people requiring transfusion", type text}, {"Serious adverse events", type text}, {"All-

```



```

1  cause mortality", type text}, {"Number of units of allogenic blood transfused", type
2  text}, {"Infection", type text}, {"Length of stay", type text}, {"Acute myocardial
3  infarction", type text}, {"Volume of allogenic blood transfused", type text}, {"Total
4  blood loss volume", type text}}, "en-GB"), {"Proportion of patients requiring
5  transfusion", "Thrombotic complications", "Duplicate outcome proportion of people
6  requiring transfusion", "Serious adverse events", "All-cause mortality", "Number of
7  units of allogenic blood transfused", "Infection", "Length of stay", "Acute myocardial
8  infarction", "Volume of allogenic blood transfused", "Total blood loss
9  volume"}, Combiner.CombineTextByDelimiter("", QuoteStyle.None), "Summed
10 weighted directness number"),
11   #"Changed Type3" = Table.TransformColumnTypes(#"Merged
12 Columns", {"Summed weighted directness number", type number}),
13   #"Calculation of directness" = Table.AddColumn(#"Changed Type3", "Outcome
14 directness", each if [Summed weighted directness number] < 0 then "?" else if
15 [Summed weighted directness number] <= 0.66 then "Not serious" else if [Summed
16 weighted directness number] <= 1.32 then "Serious" else if [Summed weighted
17 directness number] <= 2 then "Very serious" else "?"),
18   #"Analyst manual edit of directness" = Table.AddColumn(#"Calculation of
19 directness", "Edited outcome directness", each if [Protocol Outcome] = "Falls" then
20 null else null),
21   #"Directness final determination" = Table.AddColumn(#"Analyst manual edit of
22 directness", "Indirectness", each if [Edited outcome directness] <> null then [Edited
23 outcome directness] else [Outcome directness]),
24   #"Indirectness reason determination" = Table.AddColumn(#"Directness final
25 determination", "Indirectness reason", each if [Protocol Outcome] = "Physical
26 dependency" then "Indirectness: Downgraded once. Serious indirectness due to
27 >50% of overall weighting partially direct or indirect. Outcome indirect due to it
28 including mortality and physical dependency" else null),
29   #"Inserted Merged Column" = Table.AddColumn(#"Indirectness reason
30 determination", "GRADE code", each Text.Combine({Text.From([Analysis group],
31 "en-GB"), Text.From([Analysis number], "en-GB")}, "-"), type text),
32   #"Removed Columns5" = Table.RemoveColumns(#"Inserted Merged
33 Column", {"Protocol Outcome", "Scale", "Outcome short", "Summed weighted
34 directness number", "Outcome directness", "Edited outcome directness"}),

```

```

1      #"Reordered Columns2" = Table.ReorderColumns(#"Removed
2 Columns5",{ "Analysis group", "Analysis number", "GRADE code", "Indirectness",
3 "Indirectness reason"})
4 in
5      #"Reordered Columns2"
6

```

7 **Imprecision**

```

8 let
9     Source = Excel.Workbook(Web.Contents("source"), null, true),
10     #"Overall estimates_Sheet" = Source{[Item="Overall
11 estimates",Kind="Sheet"]}[Data],
12     #"Promoted Headers" = Table.PromoteHeaders(#"Overall estimates_Sheet",
13 [PromoteAllScalars=true]),
14     #"Changed Type" = Table.TransformColumnTypes(#"Promoted
15 Headers",{{"Analysis group", Int64.Type}, {"Analysis number", Int64.Type}, {"Analysis
16 name", type text}, {"Analysis group name", type text}, {"Data source", type text},
17 {"Data source eligibility", type text}, {"Data type", type text}, {"Log-scale data", type
18 logical}, {"Outcome", type text}, {"Intervention grouping", type text}, {"Experimental
19 intervention", type any}, {"Control intervention", type any}, {"Subgroup by", type any},
20 {"Filter criteria", type any}, {"Experimental group label", type text}, {"Control group
21 label", type text}, {"Statistical method", type text}, {"Effect measure", type text}, {"Unit
22 of effect measure", type any}, {"Analysis model", type text}, {"Heterogeneity
23 estimator", type any}, {"Tau² CI", type logical}, {"Subgroup estimates", type logical},
24 {"Overall estimates", type logical}, {"Test for subgroup differences", type logical},
25 {"Prediction interval", type logical}, {"Swap event and non-event", type logical}, {"CI
26 method", type text}, {"CI/PI level", type number}, {"Experimental cases", Int64.Type},
27 {"Experimental N", Int64.Type}, {"Control cases", Int64.Type}, {"Control N",
28 Int64.Type}, {"Mean", type number}, {"CI start", type number}, {"CI end", type
29 number}, {"PI start", type any}, {"PI end", type any}, {"Heterogeneity Tau²", type any},
30 {"Tau² CI start", type any}, {"Tau² CI end", type any}, {"Heterogeneity Chi²", type
31 number}, {"Heterogeneity df", Int64.Type}, {"Heterogeneity P", type number},
32 {"Heterogeneity I²", type number}, {"Effect Z", type number}, {"Effect P", type

```

```

1  number}, {"Subgroup Chi²", type any}, {"Subgroup df", type any}, {"Subgroup P", type
2  any}, {"Subgroup I²", type any}, {"ID", type text})),
3    #"Filtered Rows" = Table.SelectRows("#Changed Type", each ([Analysis group] <>
4    5 and [Analysis group] <> 6)),
5    #"Removed Other Columns" = Table.SelectColumns("#Filtered Rows",{"Analysis
6    group", "Analysis number", "Analysis name", "Analysis group name", "Outcome",
7    "Experimental intervention", "Control intervention", "Effect measure", "Experimental
8    cases", "Experimental N", "Control cases", "Control N", "Mean", "CI start", "CI end"}),
9    #"Calculating sample size" = Table.AddColumn("#Removed Other Columns",
10   "Sample size", each [Experimental N] + [Control N], Int64.Type),
11   #"Renamed Columns" = Table.RenameColumns("#Calculating sample
12   size",{"Experimental N", "n1"}, {"Control N", "n2"}, {"Experimental cases", "Number of
13   events 1"}, {"Control cases", "Number of events 2"})),
14   #"Calculating non-events 1" = Table.AddColumn("#Renamed Columns", "Non-
15   events 1", each [n1] - [Number of events 1], Int64.Type),
16   #"Calculating non-events 2" = Table.AddColumn("#Calculating non-events 1",
17   "Non-events 2", each [n2] - [Number of events 2], Int64.Type),
18   #"Calculating Risk1" = Table.AddColumn("#Calculating non-events 2", "Risk1",
19   each [Number of events 1] / [n1], type number),
20   #"Calculating Risk2" = Table.AddColumn("#Calculating Risk1", "Risk2", each
21   [Number of events 2] / [n2], type number),
22   #"Calculating risk difference" = Table.AddColumn("#Calculating Risk2", "Risk
23   difference", each [Risk1] - [Risk2], type number),
24   #"Inserted Multiplication1" = Table.AddColumn("#Calculating risk difference",
25   "Event1xNonevent1", each [Number of events 1] * [#"Non-events 1"], Int64.Type),
26   #"Inserted Multiplication2" = Table.AddColumn("#Inserted Multiplication1",
27   "EventxNonevent2", each [Number of events 2] * [#"Non-events 2"], Int64.Type),
28   #"Inserted Cube" = Table.AddColumn("#Inserted Multiplication2", "n1^3", each
29   Number.Power([#"Non-events 1"], 3), Int64.Type),
30   #"Inserted Cube1" = Table.AddColumn("#Inserted Cube", "n2^3", each
31   Number.Power([n2], 3), Int64.Type),
32   #"Inserted Division3" = Table.AddColumn("#Inserted Cube1",
33   "eventxnonevent1/n1^3", each [Event1xNonevent1] / [#"n1^3"], type number),

```

```

1      #"Inserted Division4" = Table.AddColumn(#"Inserted Division3", "Division", each
2      [EventxNonevent2] / [#"n2^3"], type number),
3      #"Renamed Columns1" = Table.RenameColumns(#"Inserted
4      Division4",{{"Division", "eventxnonevent2/n2^3"}}),
5      #"Inserted Addition2" = Table.AddColumn(#"Renamed Columns1",
6      "eventxnonevent1/n1^3 + eventxnonevent2/n2^3", each [#"eventxnonevent1/n1^3"] +
7      [#"eventxnonevent2/n2^3"], type number),
8      #"Calculate SE(RD)" = Table.AddColumn(#"Inserted Addition2", "SE(RD)", each
9      Number.Sqrt([#"eventxnonevent1/n1^3 + eventxnonevent2/n2^3"]), type number),
10     #"Inserted Multiplication3" = Table.AddColumn(#"Calculate SE(RD)", "1.96*SE",
11     each [#"SE(RD)"] * 1.96, type number),
12     #"Calculate RD 2.5% CI" = Table.AddColumn(#"Inserted Multiplication3", "RD
13     2.5% CI", each [Risk difference] - [#"1.96*SE"], type number),
14     #"Calculate RD 97.5% CI" = Table.AddColumn(#"Calculate RD 2.5% CI", "RD
15     97.5% CI", each [Risk difference] + [#"1.96*SE"], type number),
16     #"Calculating events/1000 people" = Table.AddColumn(#"Calculate RD 97.5% CI",
17     "Events/1000", each [Risk difference] * 1000, type number),
18     #"Calculating 2.5% CI" = Table.AddColumn(#"Calculating events/1000 people",
19     "Events/1000 2.5% CI", each [#"RD 2.5% CI"] * 1000, type number),
20     #"Calculating 97.5% CI" = Table.AddColumn(#"Calculating 2.5% CI", "Events/1000
21     97.5% CI", each [#"RD 97.5% CI"] * 1000, type number),
22     #"Changed Type2" = Table.TransformColumnTypes(#"Calculating 97.5%
23     CI",{{"Events/1000 97.5% CI", Int64.Type}, {"Events/1000 2.5% CI", Int64.Type},
24     {"Events/1000", Int64.Type}}),
25     #"Removed Other Columns1" = Table.SelectColumns(#"Changed
26     Type2",{ "Analysis group", "Analysis number", "Analysis name", "Analysis group
27     name", "Outcome", "Experimental intervention", "Control intervention", "Effect
28     measure", "Number of events 1", "n1", "Number of events 2", "n2", "Mean", "CI start",
29     "CI end", "Sample size", "Events/1000", "Events/1000 2.5% CI", "Events/1000 97.5%
30     CI"}),
31     #"Subtracting CIs to calculate SE" = Table.AddColumn(#"Removed Other
32     Columns1", "SE", each [CI end] - [CI start], type number),
33     #"Dividing by 3.92 to calculate SE" = Table.TransformColumns(#"Subtracting CIs
34     to calculate SE", {{"SE", each _ / 3.92, type number}}),

```

```

1      #"Square Root of sample size" = Table.AddColumn("#Dividing by 3.92 to calculate
2 SE", "Square Root(N)", each Number.Sqrt([Sample size]), type number),
3      #"Calculate SD" = Table.AddColumn("#Square Root of sample size", "SD", each
4 [#"Square Root(N)"] * [SE], type number),
5      #"SD squared" = Table.AddColumn("#Calculate SD", "SD^2", each
6 Number.Power([SD], 2), type number),
7      #"Choosing effect for MID precision calc" = Table.AddColumn("#SD squared",
8 "Events for MIDs", each if [Effect measure] = "Risk Ratio" then [Mean] else if [Effect
9 measure] = "Odds Ratio" then [Mean] else if [Effect measure] = "Risk Difference"
10 then [#"Events/1000"] else [Mean]),
11      #"Choosing 2.5% CI for MID precision calc" = Table.AddColumn("#Choosing effect
12 for MID precision calc", "2.5% CI for MIDs", each if [Effect measure] = "Risk Ratio"
13 then [CI start] else if [Effect measure] = "Odds Ratio" then [CI start] else if [Effect
14 measure] = "Risk Difference" then [#"Events/1000 2.5% CI"] else [CI start]),
15      #"Choosing 97.5% CI for MID precision calc" = Table.AddColumn("#Choosing
16 2.5% CI for MID precision calc", "97.5% CI for MIDs", each if [Effect measure] = "Risk
17 Ratio" then [CI end] else if [Effect measure] = "Odds Ratio" then [CI end] else if
18 [Effect measure] = "Risk Difference" then [#"Events/1000 97.5% CI"] else [CI end]),
19      #"Analyst inputting MIDs" = Table.AddColumn("#Choosing 97.5% CI for MID
20 precision calc", "MID", each if [Outcome] = "Person/participant generic health-related
21 quality of life - SF-36 physical component score" then 2 else if [Outcome] =
22 "Person/participant generic health-related quality of life - SF-36 mental component
23 score" then 3 else if [Effect measure] = "Risk Ratio" then 0.8 else if [Effect measure]
24 = "Hazard Ratio" then 0.8 else if [Effect measure] = "Odds Ratio" then 0.8 else if
25 [Effect measure] = "Ratio of Means" then 0.8 else null),
26      #"Calculating inverse MIDs (assuming linear scale) - for non-linear scale determine
27 differently" = Table.AddColumn("#Analyst inputting MIDs", "Inverse MID", each [MID]
28 * -1, type number),
29      #"Analyst inverse MID choice" = Table.AddColumn("#Calculating inverse MIDs
30 (assuming linear scale) - for non-linear scale determine differently", "Inverse MID
31 choice", each if [Effect measure] = "Risk Ratio" then 1.25 else if [Effect measure] =
32 "Hazard Ratio" then 1.25 else if [Effect measure] = "Odds Ratio" then 1.25 else if
33 [Effect measure] = "Ratio of Means" then 1.25 else [Inverse MID]),

```

```

1      #"2.5% CI relative precision value determination" = Table.AddColumn("#Analyst
2  inverse MID choice", "2.5% CI MID relation", each if [MID] = null then null else if
3  [#"2.5% CI for MIDs"] < [Inverse MID choice] then -1 else if [#"2.5% CI for MIDs"] <
4  [MID] then 0 else if [#"2.5% CI for MIDs"] >= [MID] then 1 else null),
5      #"97.5% CI relative precision value determination" = Table.AddColumn("#2.5% CI
6  relative precision value determination", "97.5% CI MID relation", each if [MID] = null
7  then null else if [#"97.5% CI for MIDs"] < [Inverse MID choice] then -1 else if [#"97.5%
8  CI for MIDs"] < [MID] then 0 else if [#"97.5% CI for MIDs"] >= [MID] then 1 else null),
9      #"Changed Type1" = Table.TransformColumnTypes("#97.5% CI relative precision
10 value determination",{{"2.5% CI MID relation", Int64.Type}, {"97.5% CI MID relation",
11 Int64.Type}}),
12      #"Combining precision CI values together" = Table.AddColumn("#Changed
13 Type1", "Precision number", each Text.Combine({Text.From([#"2.5% CI MID
14 relation"], "en-GB"), Text.From([#"97.5% CI MID relation"], "en-GB"}), "="), type text),
15      #"Replaced Value1" = Table.ReplaceValue("#Combining precision CI values
16 together", "", null, Replacer.ReplaceValue, {"Precision number"}),
17      #"Imprecision MID calculation" = Table.AddColumn("#Replaced Value1",
18 "Imprecision MID", each if [Precision number] = null then null else if [Precision
19 number] = "0=0" then "Not serious" else if [Precision number] = "1=1" then "Not
20 serious" else if [Precision number] = "-1=-1" then "Not serious" else if [Precision
21 number] = "0=1" then "Serious" else if [Precision number] = "1=0" then "Serious" else
22 if [Precision number] = "-1=0" then "Serious" else if [Precision number] = "0=-1" then
23 "Serious" else if [Precision number] = "-1=1" then "Very serious" else if [Precision
24 number] = "1=-1" then "Very serious" else "?"),
25      #"Merged Queries" = Table.NestedJoin("#Imprecision MID calculation", {"Analysis
26 group", "Analysis number"}, #"OIS dichotomous", {"Analysis group", "Analysis
27 number"}, "OIS dichotomous", JoinKind.LeftOuter),
28      ///If there were continuous outcomes, you would also merge the query for OIS
29 continuous, get the OIS precision for that, merge that with the OIS precision for
30 dichotomous and then do the rest of the steps as usual
31      #"Expanded OIS dichotomous" = Table.ExpandTableColumn("#Merged Queries",
32 "OIS dichotomous", {"OIS", "OIS Precision"}, {"OIS", "OIS Precision"}),

```

```

1      #"Merged Queries1" = Table.NestedJoin(#"Expanded OIS dichotomous",
2 {"Analysis group", "Analysis number"}, #"OIS continuous", {"Analysis group",
3 "Analysis number"}, "OIS continuous", JoinKind.LeftOuter),
4      #"Expanded OIS continuous" = Table.ExpandTableColumn(#"Merged Queries1",
5 "OIS continuous", {"OIS", "Precision OIS"}, {"OIS.1", "Precision OIS"}),
6      #"Merged Columns" =
7 Table.CombineColumns(Table.TransformColumnTypes(#"Expanded OIS
8 continuous", {"OIS", type text}, {"OIS.1", type text}}, "en-GB"), {"OIS",
9 "OIS.1"}, Combiner.CombineTextByDelimiter("", QuoteStyle.None), "OIS"),
10     #"Replaced Value" = Table.ReplaceValue(#"Merged Columns", "Not
11 applicable", null, Replacer.ReplaceValue, {"OIS Precision"}),
12     #"Merged Columns1" = Table.CombineColumns(#"Replaced Value", {"Precision
13 OIS", "OIS Precision"}, Combiner.CombineTextByDelimiter("", QuoteStyle.None), "OIS
14 Precision"),
15     #"Replaced Value3" = Table.ReplaceValue(#"Merged
16 Columns1", "", null, Replacer.ReplaceValue, {"OIS Precision", "OIS"}),
17     #"Analyst imprecision edit" = Table.AddColumn(#"Replaced Value3", "Imprecision
18 analyst", each if [Outcome] = "Acute myocardial infarction" then "Very serious" else
19 null),
20     #"Final imprecision calculation" = Table.AddColumn(#"Analyst imprecision edit",
21 "Imprecision", each if [Imprecision analyst] <> null then [Imprecision analyst] else if
22 [Imprecision MID] <> null then [Imprecision MID] else if [OIS Precision] <> null then
23 [OIS Precision] else null),
24     #"Method used to determine precision" = Table.AddColumn(#"Final imprecision
25 calculation", "Method used", each if [Imprecision analyst] <> null then "Imprecision
26 analyst" else if [Imprecision MID] <> null then "Imprecision MID" else if [OIS
27 Precision] <> null then "OIS Precision" else null),
28     #"Removing not serious" = Table.AddColumn(#"Method used to determine
29 precision", "Imprecision for reason", each if [Imprecision] = "Not serious" then null
30 else [Imprecision]),
31     #"Removing method for not serious" = Table.AddColumn(#"Removing not
32 serious", "Method for reason", each if [Imprecision for reason] = null then null else
33 [Method used]),

```

```

1      #"Combining method and imprecision" = Table.AddColumn(#"Removing method
2  for not serious", "Method-Imprecision", each Text.Combine({[Method for reason],
3  [Imprecision for reason]}, "-"), type text),
4      #"Replaced Value2" = Table.ReplaceValue(#"Combining method and
5  imprecision", "", null, Replacer.ReplaceValue, {"Method-Imprecision"}),
6      #"Imprecision reason generation" = Table.AddColumn(#"Replaced Value2",
7  "Imprecision reason", each if [Outcome] = "Mortality" then "Downgraded by 1 interval
8  as agreed with committee" else if [#"Method-Imprecision"] = "Imprecision MID-
9  Serious" then "Serious imprecision because 95%CI crosses 1 MID (RR 0.8-1.25)"
10 else if [#"Method-Imprecision"] = "Imprecision MID-Very serious" then "Very serious
11 imprecision because 95%CI crosses 2 MIDs (RR 0.8-1.25)" else if [#"Method-
12 Imprecision"] = "OIS Precision-Serious" then "Serious imprecision because the
13 number of participants is between 30% and 100% of the optimal information size
14 (OIS)" else if [#"Method-Imprecision"] = "OIS Precision-Very serious" then "Very
15 serious imprecision because the number of participants is less than 30% of the
16 optimal information size (OIS) or the ratio between the confidence intervals exceed
17 2.5 (odds ratio)/3 (risk ratio) or both" else null),
18      #"Removed Other Columns2" = Table.SelectColumns(#"Imprecision reason
19 generation", {"Analysis group", "Analysis number", "Analysis name", "Number of
20 events 1", "n1", "Number of events 2", "n2", "Sample size", "Events/1000",
21 "Events/1000 2.5% CI", "Events/1000 97.5% CI", "SE", "Square Root(N)", "SD",
22 "SD^2", "Events for MIDs", "2.5% CI for MIDs", "97.5% CI for MIDs", "Imprecision",
23 "Imprecision reason"}),
24      #"Inserted Merged Column" = Table.AddColumn(#"Removed Other Columns2",
25 "GRADE group", each Text.Combine({Text.From([Analysis group], "en-GB"),
26 Text.From([Analysis number], "en-GB")}, "-"), type text),
27      #"Filtered Rows1" = Table.SelectRows(#"Inserted Merged Column", each
28 ([Analysis group] = 1))
29 in
30      #"Filtered Rows1"
31

```

32 Number of studies

33 let


```

1     Source = Excel.Workbook(Web.Contents("source"), null, true),
2     #"Data rows_Sheet" = Source[[Item="Data rows",Kind="Sheet"]][Data],
3     #"Promoted Headers" = Table.PromoteHeaders(#"Data rows_Sheet",
4 [PromoteAllScalars=true]),
5     #"Changed Type" = Table.TransformColumnTypes(#"Promoted
6 Headers",{{"Analysis group", Int64.Type}, {"Analysis number", Int64.Type}, {"Analysis
7 name", type text}, {"Subgroup", type any}, {"Applicability", type text}, {"Study", type
8 text}, {"Study year", Int64.Type}, {"GIV Mean", type number}, {"GIV SE", type
9 number}, {"Experimental mean", type number}, {"Experimental SD", type number},
10 {"Experimental cases", Int64.Type}, {"Experimental N", Int64.Type}, {"Control mean",
11 type number}, {"Control SD", type number}, {"Control cases", Int64.Type}, {"Control
12 N", Int64.Type}, {"O-E", type any}, {"Variance", type any}, {"Weight", type number},
13 {"Mean", type number}, {"CI start", type number}, {"CI end", type number},
14 {"Footnotes", type any}}),
15     #"Splitting by "" at "" to separate follow up time" = Table.SplitColumn(#"Changed
16 Type", "Analysis name", Splitter.SplitTextByDelimiter(" at ", QuoteStyle.Csv),
17 {"Analysis name.1", "Analysis name.2"}),
18     #"Changed Type1" = Table.TransformColumnTypes(#"Splitting by "" at "" to
19 separate follow up time",{{"Analysis name.1", type text}, {"Analysis name.2", type
20 text}}),
21     #"Renamed Columns" = Table.RenameColumns(#"Changed Type1",{{"Analysis
22 name.2", "Follow up time"}}),
23     #"Splitting by "" (" to separate scale information" = Table.SplitColumn(#"Renamed
24 Columns", "Analysis name.1", Splitter.SplitTextByDelimiter(" (", QuoteStyle.Csv),
25 {"Analysis name.1.1", "Analysis name.1.2"}),
26     #"Changed Type2" = Table.TransformColumnTypes(#"Splitting by "" (" to
27 separate scale information",{{"Analysis name.1.1", type text}, {"Analysis name.1.2",
28 type text}}),
29     #"Renamed Columns1" = Table.RenameColumns(#"Changed Type2",{{"Analysis
30 name.1.2", "Scale"}, {"Analysis name.1.1", "Protocol Outcome"}}),
31     #"Removing closing bracket" = Table.ReplaceValue(#"Renamed
32 Columns1",")", "", Replacer.ReplaceText,{"Scale"}),
33     #"Making short outcome" = Table.AddColumn(#"Removing closing bracket",
34 "Outcome short", each if [Protocol Outcome] = "Mortality" then "Mort" else if [Protocol

```

```

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

```

```

1      #"Inserted Merged Column" = Table.AddColumn(#"Inserted Sum", "GRADE
2 group", each Text.Combine({Text.From([Analysis group], "en-GB"),
3 Text.From([Analysis number], "en-GB")}, "-"), type text),
4      #"Removed Columns1" = Table.RemoveColumns(#"Inserted Merged
5 Column",{ "Proportion of patients requiring transfusion", "Thrombotic complications",
6 "Duplicate outcome proportion of people requiring transfusion", "Serious adverse
7 events", "All-cause mortality", "Number of units of allogenic blood transfused",
8 "Infection", "Length of stay", "Acute myocardial infarction", "Volume of allogenic blood
9 transfused", "Total blood loss volume"})
10 in
11      #"Removed Columns1"
12

```

13 **Timepoint**

```

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

```

```

1   #"Changed Type1" = Table.TransformColumnTypes(#"Splitting by "" at "" to
2   separate follow up time",{{"Analysis name.1", type text}, {"Analysis name.2", type
3   text}}),
4   #"Renamed Columns" = Table.RenameColumns(#"Changed Type1",{{"Analysis
5   name.2", "Follow up time"}}),
6   #"Splitting by "" (" to separate scale information" = Table.SplitColumn(#"Renamed
7   Columns", "Analysis name.1", Splitter.SplitTextByDelimiter(" (", QuoteStyle.Csv),
8   {"Analysis name.1.1", "Analysis name.1.2"}),
9   #"Changed Type2" = Table.TransformColumnTypes(#"Splitting by "" (" to
10  separate scale information",{{"Analysis name.1.1", type text}, {"Analysis name.1.2",
11  type text}}),
12  #"Renamed Columns1" = Table.RenameColumns(#"Changed Type2",{{"Analysis
13  name.1.2", "Scale"}, {"Analysis name.1.1", "Protocol Outcome"}}),
14  #"Removing closing bracket" = Table.ReplaceValue(#"Renamed
15  Columns1",")","",Replacer.ReplaceText,{"Scale"}),
16  #"Making short outcome" = Table.AddColumn(#"Removing closing bracket",
17  "Outcome short", each if [Protocol Outcome] = "Mortality" then "Mort" else if [Protocol
18  Outcome] = "Physical dependency" then "Pdep" else if [Protocol Outcome] = "Falls"
19  then "Falls" else if [Protocol Outcome] = "Readmissions to hospital" then "Read" else
20  if [Protocol Outcome] = "Person/participant generic health-related quality of life" then
21  "HQOL" else if [Protocol Outcome] = "Activities of daily living" then "ADL" else if
22  [Protocol Outcome] = "Extended activities of daily living" then "EADL" else if [Protocol
23  Outcome] = "Length of hospital stay" then "LOS" else if [Protocol Outcome] =
24  "Caregiver strain index" then "CSI" else if [Protocol Outcome] = "Psychological
25  distress/mood" then "PD" else if [Protocol Outcome] = "Carer generic health-related
26  quality of life" then "CHQOL" else if [Protocol Outcome] = "Stroke-specific Patient-
27  Reported Outcome Measures" then "SQOL" else "?"),
28  #"Merged Queries" = Table.NestedJoin(#"Making short outcome", {"Study"},
29  BiasInd, {"Study"}, "Critical appraisal no edits", JoinKind.LeftOuter),
30  #"Expanded Critical appraisal no edits" = Table.ExpandTableColumn(#"Merged
31  Queries", "Critical appraisal no edits", {"Overall risk of bias", "Overall directness",
32  "Short population", "Short comparison"}, {"Overall risk of bias", "Overall directness",
33  "Short population", "Short comparison"}),
34  #"Removed Duplicates" = Table.Distinct(#"Expanded Critical appraisal no edits"),

```

```

1      #"Removed Columns" = Table.RemoveColumns(#"Removed
2 Duplicates",{ "Footnotes", "GIV Mean", "GIV SE", "Experimental mean", "Experimental
3 SD", "Experimental cases", "Experimental N", "Control mean", "Control SD", "Control
4 cases", "Control N", "O-E", "Variance", "Mean", "CI start", "CI end"}),
5      #"Reordered Columns" = Table.ReorderColumns(#"Removed Columns",{ "Analysis
6 group", "Analysis number", "Protocol Outcome", "Scale", "Outcome short", "Short
7 population", "Short comparison", "Follow up time", "Subgroup", "Applicability",
8 "Study", "Study year", "Weight", "Overall risk of bias", "Overall directness"}),
9      #"Removed Columns1" = Table.RemoveColumns(#"Reordered Columns",{ "Follow
10 up time"}),
11      #"Reordered Columns1" = Table.ReorderColumns(#"Removed
12 Columns1",{ "Analysis group", "Analysis number", "Study", "Protocol Outcome",
13 "Scale", "Outcome short", "Short population", "Short comparison", "Subgroup",
14 "Applicability", "Study year", "Weight", "Overall risk of bias", "Overall directness"}),
15      #"Removed Columns2" = Table.RemoveColumns(#"Reordered Columns1",{ "Study
16 year"}),
17      #"Removed Columns3" = Table.RemoveColumns(#"Removed
18 Columns2",{ "Study", "Short population", "Short comparison", "Subgroup",
19 "Applicability"}),
20      #"Removed Columns4" = Table.RemoveColumns(#"Removed
21 Columns3",{ "Weight", "Overall risk of bias", "Overall directness"}),
22      #"Duplicated Column" = Table.DuplicateColumn(#"Removed Columns4", "Protocol
23 Outcome", "Protocol Outcome - Copy"),
24      #"Added Custom" = Table.AddColumn(#"Duplicated Column", "Timepoint
25 (months)", each 0),
26      #"Pivoted Column to calculate average timepoint (months)" = Table.Pivot(#"Added
27 Custom", List.Distinct(#"Added Custom"[#"Protocol Outcome - Copy"]), "Protocol
28 Outcome - Copy", "Timepoint (months)", List.Average),
29      #"Merged Columns" =
30 Table.CombineColumns(Table.TransformColumnTypes(#"Pivoted Column to
31 calculate average timepoint (months)", {{"Proportion of patients requiring
32 transfusion", type text}, {"Thrombotic complications", type text}, {"Duplicate outcome
33 proportion of people requiring transfusion", type text}, {"Serious adverse events", type
34 text}, {"All-cause mortality", type text}, {"Number of units of allogenic blood

```

```

1 transfused", type text}, {"Infection", type text}, {"Length of stay", type text}, {"Acute
2 myocardial infarction", type text}, {"Volume of allogenic blood transfused", type text},
3 {"Total blood loss volume", type text}}, "en-GB"), {"Proportion of patients requiring
4 transfusion", "Thrombotic complications", "Duplicate outcome proportion of people
5 requiring transfusion", "Serious adverse events", "All-cause mortality", "Number of
6 units of allogenic blood transfused", "Infection", "Length of stay", "Acute myocardial
7 infarction", "Volume of allogenic blood transfused", "Total blood loss
8 volume"}, Combiner.CombineTextByDelimiter("", QuoteStyle.None), "Average
9 timepoint (months)",
10     #"Changed Type4" = Table.TransformColumnTypes(#"Merged
11 Columns", {"Average timepoint (months)", type number}),
12     #"Rounded Off" = Table.TransformColumns(#"Changed Type4", {"Average
13 timepoint (months)", each Number.Round(_, 1), type number}),
14     #"Inserted Merged Column" = Table.AddColumn(#"Rounded Off", "GRADE code",
15     each Text.Combine({Text.From([Analysis group], "en-GB"), Text.From([Analysis
16 number], "en-GB")}, "-"), type text)
17 in
18     #"Inserted Merged Column"
19

```

20 Inconsistency, units

```

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

```

```

1  label", type text}, {"Statistical method", type text}, {"Effect measure", type text}, {"Unit
2  of effect measure", type any}, {"Analysis model", type text}, {"Heterogeneity
3  estimator", type any}, {"Tau2 CI", type logical}, {"Subgroup estimates", type logical},
4  {"Overall estimates", type logical}, {"Test for subgroup differences", type logical},
5  {"Prediction interval", type logical}, {"Swap event and non-event", type logical}, {"CI
6  method", type text}, {"CI/PI level", type number}, {"Experimental cases", Int64.Type},
7  {"Experimental N", Int64.Type}, {"Control cases", Int64.Type}, {"Control N",
8  Int64.Type}, {"Mean", type number}, {"CI start", type number}, {"CI end", type
9  number}, {"PI start", type any}, {"PI end", type any}, {"Heterogeneity Tau2", type any},
10 {"Tau2 CI start", type any}, {"Tau2 CI end", type any}, {"Heterogeneity Chi2", type
11 number}, {"Heterogeneity df", Int64.Type}, {"Heterogeneity P", type number},
12 {"Heterogeneity I2", type number}, {"Effect Z", type number}, {"Effect P", type
13 number}, {"Subgroup Chi2", type any}, {"Subgroup df", type any}, {"Subgroup P", type
14 any}, {"Subgroup I2", type any}, {"ID", type text})),
15   #"Removed Columns" = Table.RemoveColumns(#"Changed Type",{ "PI start", "PI
16   end", "Heterogeneity Tau2", "Tau2 CI start", "Tau2 CI end", "Heterogeneity Chi2",
17   "Heterogeneity df", "Heterogeneity P", "Effect Z", "Effect P", "Subgroup Chi2",
18   "Subgroup df", "Subgroup P", "ID", "CI method", "Test for subgroup differences",
19   "Prediction interval", "Swap event and non-event", "Tau2 CI", "Heterogeneity
20   estimator", "Experimental intervention", "Control intervention"})),
21   #"Inconsistency calculation" = Table.AddColumn(#"Removed Columns",
22   "Inconsistency calculation", each if [#"Heterogeneity I2"] = null then "Serious - single
23   study?" else if [#"Heterogeneity I2"] < 0 then "?" else if [#"Heterogeneity I2"] < 40 then
24   "Not serious" else if [#"Heterogeneity I2"] < 60 then "Serious" else if [#"Heterogeneity
25   I2"] < 80 then "Very serious" else if [#"Heterogeneity I2"] < 100 then "Very serious -
26   too heterogenous to meaningfully pool?" else "?"),
27   #"Analyst edit of inconsistency" = Table.AddColumn(#"Inconsistency calculation",
28   "Inconsistency analyst check", each if [Analysis name] = "Volume of allogenic blood
29   transfused (mL) at end of study" then "Serious" else if [Analysis name] = "Acute
30   myocardial infarction at end of study" then "Serious" else if [Analysis name] = "Length
31   of stay (days) at end of study" then "Very serious" else if [Analysis name] = "Total
32   blood loss volume at end of study" then "Very serious" else null),

```

```

1      #"Final inconsistency calculation" = Table.AddColumn(#"Analyst edit of
2      inconsistency", "Inconsistency", each if [Inconsistency analyst check] <> null then
3      [Inconsistency analyst check] else [Inconsistency calculation]),
4      #"Inconsistency reason" = Table.AddColumn(#"Final inconsistency calculation",
5      "Inconsistency reason", each if Text.Contains([Inconsistency calculation], "single
6      study") then "Inconsistency: Single study- downgraded once for inconsistency, as
7      single study outcomes may otherwise receive favourable ratings for inconsistency by
8      default" else if [#"Heterogeneity I2"] < 40 then null else if [#"Heterogeneity I2"] < 60
9      then "Inconsistency: Downgraded once. Serious heterogeneity (I2 = 40 to 60%)
10     unexplained by subgroup analysis. Random effects analysis used" else if
11     [#"Heterogeneity I2"] < 80 then "Inconsistency: Downgraded twice. Very serious
12     heterogeneity (serious I2 = >60%) unexplained by subgroup analysis. Random
13     effects analysis used" else if [#"Heterogeneity I2"] <= 100 then "Inconsistency:
14     Downgraded twice. Very serious heterogeneity (serious I2 = >60%) unexplained by
15     subgroup analysis. Random effects analysis used" else if [#"Heterogeneity I2"] > 100
16     then "?" else if [Analysis name] = "Mortality" then null else null),
17     #"Removed Columns1" = Table.RemoveColumns(#"Inconsistency
18     reason",{"Inconsistency calculation", "Inconsistency analyst check"}),
19     #"Inserted Merged Column" = Table.AddColumn(#"Removed Columns1", "GRADE
20     group", each Text.Combine({Text.From([Analysis group], "en-GB"),
21     Text.From([Analysis number], "en-GB")}, "-"), type text),
22     #"Analyst editable units column" = Table.AddColumn(#"Inserted Merged Column",
23     "Units", each if [Unit of effect measure] <> null then [Unit of effect measure] else if
24     [Outcome] = "Length of stay" then "days" else if [Outcome] = "Number of units of
25     allogenic blood transfused" then "units" else if [Outcome] = "Volume of allogenic
26     blood transfused" then "mL" else "Not applicable"),
27     #"Added Custom1" = Table.AddColumn(#"Analyst editable units column", "|", each
28     "|"),
29     #"Analyst editable study type" = Table.AddColumn(#"Added Custom1", "Study
30     type", each if [Outcome] = "a" then "Non-randomised studies" else if [Outcome] = "b"
31     then "Randomised trials" else "Randomised trials"),
32     #"Analyst editable ROBINS-I used?" = Table.AddColumn(#"Analyst editable study
33     type", "ROBINS-I used?", each if [Study type] = "Randomised trials" then false else if
34     [Outcome] = "a" then true else if [Outcome] = "b" then false else true),

```



```

1      #"Analyst editable publication bias" = Table.AddColumn(#"Analyst editable
2  ROBINS-I used?", "Publication bias", each if [Outcome] = "a" then "Undetected" else
3  if [Outcome] = "b" then "Strongly suspected" else "Undetected"),
4      #"Analyst editable large effect" = Table.AddColumn(#"Analyst editable publication
5  bias", "Large effect", each if [Study type] = "Randomised trials" then "No" else if
6  [Outcome] = "a" then "No" else if [Outcome] = "b" then "Large" else if [Outcome] = "c"
7  then "Very large" else "No"),
8      #"Analyst editable plausible confounding" = Table.AddColumn(#"Analyst editable
9  large effect", "Plausible confounding", each if [Study type] = "Randomised trials" then
10 "No" else if [Outcome] = "a" then "No" else if [Outcome] = "b" then "Would reduce
11 demonstrated effect" else if [Outcome] = "c" then "Would suggest spurious effect"
12 else "No"),
13      #"Analyst editable dose response gradient" = Table.AddColumn(#"Analyst editable
14 plausible confounding", "Dose response gradient", each if [Study type] =
15 "Randomised trials" then "No" else if [Outcome] = "a" then "No" else if [Outcome] =
16 "b" then "No" else "No"),
17      #"Making other considerations column" = Table.AddColumn(#"Analyst editable
18 dose response gradient", "Other considerations", each Text.Combine({[Publication
19 bias], [Large effect], [Plausible confounding], [Dose response gradient]}, "
20 "), type text),
21      #"Replaced Value" = Table.ReplaceValue(#"Making other considerations
22 column","Undetected
23 No
24 No
25 No","None",Replacer.ReplaceText,{"Other considerations"}),
26      #"Analyst editable importance" = Table.AddColumn(#"Replaced Value",
27 "Importance", each if [Outcome] = "Health-related quality of life" then "Critical" else if
28 [Outcome] = "Mortality" then "Critical" else if [Outcome] = "a" then "Critical" else if
29 [Outcome] = "b" then "Important" else "Critical"),
30      #"Removed Columns2" = Table.RemoveColumns(#"Analyst editable
31 importance",{[Publication bias], "Large effect", "Plausible confounding", "Dose
32 response gradient"})
33 in
34      #"Removed Columns2"

```

1

2 **Median baseline rate**

3 let

```

4     Source = Excel.Workbook(Web.Contents("source"), null, true),
5     #"Data rows_Sheet" = Source[Item="Data rows",Kind="Sheet"][Data],
6     #"Promoted Headers" = Table.PromoteHeaders(#"Data rows_Sheet",
7 [PromoteAllScalars=true]),
8     #"Changed Type" = Table.TransformColumnTypes(#"Promoted
9 Headers",{{"Analysis group", Int64.Type}, {"Analysis number", Int64.Type}, {"Analysis
10 name", type text}, {"Subgroup", type any}, {"Applicability", type text}, {"Study", type
11 text}, {"Study year", Int64.Type}, {"GIV Mean", type number}, {"GIV SE", type
12 number}, {"Experimental mean", type number}, {"Experimental SD", type number},
13 {"Experimental cases", Int64.Type}, {"Experimental N", Int64.Type}, {"Control mean",
14 type number}, {"Control SD", type number}, {"Control cases", Int64.Type}, {"Control
15 N", Int64.Type}, {"O-E", type any}, {"Variance", type any}, {"Weight", type number},
16 {"Mean", type number}, {"CI start", type number}, {"CI end", type number},
17 {"Footnotes", type any}}),
18     #"Splitting by "" at "" to separate follow up time" = Table.SplitColumn(#"Changed
19 Type", "Analysis name", Splitter.SplitTextByDelimiter(" at ", QuoteStyle.Csv),
20 {"Analysis name.1", "Analysis name.2"}),
21     #"Changed Type1" = Table.TransformColumnTypes(#"Splitting by "" at "" to
22 separate follow up time",{{"Analysis name.1", type text}, {"Analysis name.2", type
23 text}}),
24     #"Renamed Columns" = Table.RenameColumns(#"Changed Type1",{{"Analysis
25 name.2", "Follow up time"}}),
26     #"Splitting by "" (" to separate scale information" = Table.SplitColumn(#"Renamed
27 Columns", "Analysis name.1", Splitter.SplitTextByDelimiter(" (", QuoteStyle.Csv),
28 {"Analysis name.1.1", "Analysis name.1.2"}),
29     #"Changed Type2" = Table.TransformColumnTypes(#"Splitting by "" (" to
30 separate scale information",{{"Analysis name.1.1", type text}, {"Analysis name.1.2",
31 type text}}),
32     #"Renamed Columns1" = Table.RenameColumns(#"Changed Type2",{{"Analysis
33 name.1.2", "Scale"}, {"Analysis name.1.1", "Protocol Outcome"}}),

```

```

1      #"Removing closing bracket" = Table.ReplaceValue(#"Renamed
2 Columns1", ",", "", Replacer.ReplaceText, {"Scale"}),
3      #"Removed Duplicates" = Table.Distinct(#"Removing closing bracket"),
4      #"Inserted Division" = Table.AddColumn(#"Removed Duplicates", "Control group
5 rate", each [Control cases] / [Control N], type number),
6      #"Removed Columns" = Table.RemoveColumns(#"Inserted Division", {"Footnotes",
7 "O-E", "Variance", "Weight", "Mean", "CI start", "CI end", "GIV Mean", "GIV SE",
8 "Experimental mean", "Experimental SD", "Experimental cases", "Experimental N",
9 "Control mean", "Control SD", "Study year", "Study", "Control cases", "Control N"}),
10     #"Duplicated Column" = Table.DuplicateColumn(#"Removed Columns", "Protocol
11 Outcome", "Protocol Outcome - Copy"),
12     #"Pivoted Column" = Table.Pivot(#"Duplicated Column", List.Distinct(#"Duplicated
13 Column"[#"Protocol Outcome - Copy"]), "Protocol Outcome - Copy", "Control group
14 rate", List.Average),
15     #"Combining median control events" =
16 Table.CombineColumns(Table.TransformColumnTypes(#"Pivoted Column",
17 {"Proportion of patients requiring transfusion", type text}, {"All-cause mortality", type
18 text}, {"Length of stay", type text}, {"Number of units of allogenic blood transfused",
19 type text}, {"Volume of allogenic blood transfused", type text}, {"Serious adverse
20 events", type text}, {"Thrombotic complications", type text}, {"Acute myocardial
21 infarction", type text}, {"Infection", type text}, {"Duplicate outcome proportion of
22 people requiring transfusion", type text}, {"Total blood loss volume", type text}}, "en-
23 GB"), {"Proportion of patients requiring transfusion", "All-cause mortality", "Length of
24 stay", "Number of units of allogenic blood transfused", "Volume of allogenic blood
25 transfused", "Serious adverse events", "Thrombotic complications", "Acute
26 myocardial infarction", "Infection", "Duplicate outcome proportion of people requiring
27 transfusion", "Total blood loss volume"}, Combiner.CombineTextByDelimiter("",
28 QuoteStyle.None), "Median control group rate"),
29     #"Changed Type3" = Table.TransformColumnTypes(#"Combining median control
30 events", {"Median control group rate", type number}),
31     #"Producing GRADE number" = Table.AddColumn(#"Changed Type3", "GRADE
32 number", each Text.Combine({Text.From([Analysis group], "en-GB"),
33 Text.From([Analysis number], "en-GB")}, "-"), type text)
34 in

```

1 #"Producing GRADE number"

2

3 **OIS continuous**

4 let

5 Source = Excel.Workbook(Web.Contents("source"), null, true),

6 #"Overall estimates_Sheet" = Source[{Item="Data rows",Kind="Sheet"}][Data],

7 #"Promoted Headers" = Table.PromoteHeaders(#"Overall estimates_Sheet",

8 [PromoteAllScalars=true]),

9 #"Changed Type" = Table.TransformColumnTypes(#"Promoted

10 Headers",{{"Analysis group", Int64.Type}, {"Analysis number", Int64.Type}, {"Analysis

11 name", type text}, {"Study year", Int64.Type}, {"GIV Mean", type number}, {"GIV SE",

12 type number}, {"Experimental mean", type number}, {"Experimental SD", type

13 number}, {"Experimental cases", Int64.Type}, {"Experimental N", Int64.Type},

14 {"Control mean", type number}, {"Control SD", type number}, {"Control cases",

15 Int64.Type}, {"Control N", Int64.Type}, {"O-E", type any}, {"Variance", type any},

16 {"Weight", type number}, {"Mean", type number}, {"CI start", type number}, {"CI end",

17 type number}, {"Footnotes", type any}, {"Study", type text}, {"Applicability", type text},

18 {"Subgroup number", type text}, {"Subgroup", type text}}),

19 #"Removed Other Columns" = Table.SelectColumns(#"Changed Type",{ "Analysis

20 group", "Analysis number", "Analysis name", "Subgroup", "Subgroup number",

21 "Applicability", "Study", "Study year", "GIV Mean", "GIV SE", "Experimental mean",

22 "Experimental SD", "Experimental cases", "Experimental N", "Control mean", "Control

23 SD", "Control cases", "Control N", "O-E", "Variance", "Weight", "Mean", "CI start", "CI

24 end"}),

25 #"Filtered Rows" = Table.SelectRows(#"Removed Other Columns", each

26 ([Experimental mean] <> null) and ([Analysis name] <> "Total blood loss volume at

27 end of study")),

28 #"1" = Table.AddColumn(#"Filtered Rows", "1", each 1),

29 #"Replace weight value ""0"" with ""100"" =

30 Table.ReplaceValue(#"1",0,100,Replacer.ReplaceValue,{"Weight"}),

31 #"Reciprocal experimental n" = Table.AddColumn(#"Replace weight value ""0""

32 with ""100"", "1/n1", each [1] / [Experimental N], type number),

```

1      #"Reciprocal control n" = Table.AddColumn(#"Reciprocal experimental n", "1/n2",
2  each [1] / [Control N], type number),
3      #"Adding reciprocal n" = Table.AddColumn(#"Reciprocal control n", "1/n1+1/n2",
4  each [#"1/n1"] + [#"1/n2"], type number),
5      #"Calculated Square Root" = Table.TransformColumns(#"Adding reciprocal
6  n",{{"1/n1+1/n2", Number.Sqrt, type number}}),
7      #"Calculate GIV SD by dividing SE by SQRT reciprocal n" =
8  Table.AddColumn(#"Calculated Square Root", "SD (GIV)", each [GIV SE] /
9  [#"1/n1+1/n2"], type number),
10     #"Add the SDs" = Table.AddColumn(#"Calculate GIV SD by dividing SE by SQRT
11  reciprocal n", "SD (general)", each [Experimental SD] + [Control SD], type number),
12     #"Divide to work out the average SD" = Table.TransformColumns(#"Add the SDs",
13  {"SD (general)", each _ / 2, type number})),
14     #"Condition decide which SD to use" = Table.AddColumn(#"Divide to work out the
15  average SD", "Common SD", each if [Experimental SD] = 0 then [#"SD (GIV)"] else
16  [#"SD (general)"]),
17     #"Changed Type1" = Table.TransformColumnTypes(#"Condition decide which SD
18  to use",{{"Common SD", type number}}),
19     #"Calculate mean difference" = Table.AddColumn(#"Changed Type1", "Mean
20  difference calc", each [Experimental mean] - [Control mean], type number),
21     #"Condition decide which MD to use" = Table.AddColumn(#"Calculate mean
22  difference", "MD", each if [Experimental SD] = 0 then [GIV Mean] else [Mean
23  difference calc]),
24     #"Calculate SMD" = Table.AddColumn(#"Condition decide which MD to use",
25  "SMD", each [MD] / [Common SD], type number),
26     #"sqrt(2)" = Table.AddColumn(#"Calculate SMD", "sqrt(2)", each Number.Sqrt(2),
27  type number),
28     #"SD * sqrt(2)" = Table.AddColumn(#"sqrt(2)", "Bottom", each [Common SD] *
29  [#"sqrt(2)"], type number),
30     #"Calculate SMD prep" = Table.AddColumn(#"SD * sqrt(2)", "SMDprep", each
31  [MD] / [Bottom], type number),
32     #"Divided Column" = Table.TransformColumns(#"Calculate SMD prep", {"Weight",
33  each _ / 100, type number})),

```

```

1      #"Calculate weighted SMD prep" = Table.AddColumn(#"Divided Column",
2      "Weighted SMDprep", each [Weight] * [SMDprep], type number),
3      #"Removed Other Columns1" = Table.SelectColumns(#"Calculate weighted SMD
4      prep",{ "Analysis group", "Analysis number", "Analysis name", "Subgroup", "Subgroup
5      number", "Experimental N", "Control N", "Weighted SMDprep"}),
6      #"Unpivoted Columns" = Table.UnpivotOtherColumns(#"Removed Other
7      Columns1", {"Analysis group", "Analysis number", "Analysis name", "Subgroup",
8      "Subgroup number"}, "Attribute", "Value"),
9      #"Pivoting using sum to get SMD for outcome" = Table.Pivot(#"Unpivoted
10     Columns", List.Distinct(#"Unpivoted Columns"[Attribute]), "Attribute", "Value",
11     List.Sum),
12     #"Absolute value only for SMD" = Table.TransformColumns(#"Pivoting using sum
13     to get SMD for outcome",{{"Weighted SMDprep", Number.Abs, type number}}),
14     #"Rename to nonctr" = Table.RenameColumns(#"Absolute value only for
15     SMD",{{"Weighted SMDprep", "nonctr"}}),
16     #"za (assuming power of 0.8)" = Table.AddColumn(#"Rename to nonctr", "za",
17     each 0.841621234),
18     #"zb (assuming 2 tailed test, alpha of 0.05)" = Table.AddColumn(#"za (assuming
19     power of 0.8)", "zb", each 1.959963985),
20     #"za+zb" = Table.AddColumn(#"zb (assuming 2 tailed test, alpha of 0.05)",
21     "za+zb", each [za] + [zb], type number),
22     #"Divide z complex by nonctr" = Table.AddColumn(#"za+zb", "sss", each
23     [#"za+zb"] / [nonctr], type number),
24     Squared = Table.AddColumn(#"Divide z complex by nonctr", "OIS", each
25     Number.Power([sss], 2), type number),
26     #"Rounded to make OIS" = Table.TransformColumns(Squared,{"OIS", each
27     Number.Round(_, 0), type number}},
28     #"Multiply by 2 to get OIS for both arms" = Table.TransformColumns(#"Rounded to
29     make OIS", {"OIS", each _ * 2, type number}},
30     #"Calculate total N" = Table.AddColumn(#"Multiply by 2 to get OIS for both arms",
31     "Total N", each [Experimental N] + [Control N], type number),
32     #"Inserted Division" = Table.AddColumn(#"Calculate total N", "Prop OIS", each
33     [Total N] / [OIS], type number),

```

```

1      #"Added Conditional Column" = Table.AddColumn(#"Inserted Division",
2      "Precision", each if [Prop OIS] >= 1 then "No concerns regarding optimal information
3      size" else if [Total N] >= 800 then "No concerns regarding optimal information size"
4      else if [Prop OIS] >= 0.3 then "Serious concerns regarding optimal information size"
5      else if [Prop OIS] < 0.3 then "Very serious concerns regarding optimal information
6      size" else "?"),
7      #"Added Conditional Column1" = Table.AddColumn(#"Added Conditional Column",
8      "Precision OIS", each if [Prop OIS] >= 1 then "Not serious" else if [Total N] >= 800
9      then "Not serious" else if [Prop OIS] >= 0.3 then "Serious" else if [Prop OIS] < 0.3
10     then "Very serious" else "Not serious"),
11     #"Removed Other Columns2" = Table.SelectColumns(#"Added Conditional
12     Column1",{"Analysis group", "Analysis number", "Analysis name", "Subgroup",
13     "Subgroup number", "Experimental N", "Control N", "OIS", "Total N", "Precision",
14     "Precision OIS"})
15 in
16     #"Removed Other Columns2"
17

```

18 OIS dichotomous

```

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

```

```

1 estimator", type any}, {"Tau2 CI", type logical}, {"Subgroup estimates", type logical},
2 {"Overall estimates", type logical}, {"Test for subgroup differences", type logical},
3 {"Prediction interval", type logical}, {"Swap event and non-event", type logical}, {"CI
4 method", type text}, {"CI/PI level", type number}, {"Experimental cases", Int64.Type},
5 {"Experimental N", Int64.Type}, {"Control cases", Int64.Type}, {"Control N",
6 Int64.Type}, {"Mean", type number}, {"CI start", type number}, {"CI end", type
7 number}, {"PI start", type any}, {"PI end", type any}, {"Heterogeneity Tau2", type any},
8 {"Tau2 CI start", type any}, {"Tau2 CI end", type any}, {"Heterogeneity Chi2", type
9 number}, {"Heterogeneity df", Int64.Type}, {"Heterogeneity P", type number},
10 {"Heterogeneity I2", type number}, {"Effect Z", type number}, {"Effect P", type
11 number}, {"Subgroup Chi2", type any}, {"Subgroup df", type any}, {"Subgroup P", type
12 any}, {"Subgroup I2", type any}, {"ID", type text})),
13   #"Removed Other Columns" = Table.SelectColumns(#"Changed Type",{"Analysis
14 group", "Analysis number", "Analysis name", "Analysis group name", "Outcome",
15 "Intervention grouping", "Experimental intervention", "Control intervention",
16 "Experimental group label", "Control group label", "Effect measure", "Experimental
17 cases", "Experimental N", "Control cases", "Control N", "Mean", "CI start", "CI end",
18 "ID"}),
19   #"Removing mean difference and std. mean difference" =
20 Table.SelectRows(#"Removed Other Columns", each [Effect measure] <> "Mean
21 Difference" and [Effect measure] <> "Std. Mean Difference"),
22   #"Reordered Columns" = Table.ReorderColumns(#"Removing mean difference
23 and std. mean difference",{"Analysis group", "Analysis number", "ID", "Analysis
24 name", "Analysis group name", "Outcome", "Intervention grouping", "Experimental
25 intervention", "Control intervention", "Experimental group label", "Control group label",
26 "Effect measure", "Experimental cases", "Experimental N", "Control cases", "Control
27 N"}),
28   #"Calculate proportion experimental group" = Table.AddColumn(#"Reordered
29 Columns", "p1", each [Experimental cases] / [Experimental N], type number),
30   #"Calculate proportion control group" = Table.AddColumn(#"Calculate proportion
31 experimental group", "p2", each [Control cases] / [Control N], type number),
32   #"Proportion unexposed with outcomes for GIV" = Table.AddColumn(#"Calculate
33 proportion control group", "prop unexposed", each 0.5),

```



```

1      #"Risk ratios only GIV" = Table.AddColumn(#"Proportion unexposed with
2 outcomes for GIV", "Risk ratio", each if [Effect measure] = "Risk Ratio" then [Mean]
3 else null),
4      #"Odds ratios only GIV" = Table.AddColumn(#"Risk ratios only GIV", "Odds ratio",
5 each if [Effect measure] = "Odds Ratio" then [Mean] else null),
6      #"Risk differences only GIV" = Table.AddColumn(#"Odds ratios only GIV", "Risk
7 difference", each if [Effect measure] = "Risk difference" then [Mean] else null),
8      #"Changed Type2" = Table.TransformColumnTypes(#"Risk differences only
9 GIV",{{"prop unexposed", type number}, {"Risk ratio", type number}, {"Odds ratio",
10 type number}, {"Risk difference", type number}}),
11      #"OR-1" = Table.AddColumn(#"Changed Type2", "OR-1", each [Odds ratio] - 1,
12 type number),
13      #"proportion unexposed+1" = Table.AddColumn(#"OR-1", "1+pe", each [prop
14 unexposed] + 1, type number),
15      #"Top equation portion" = Table.AddColumn(#"proportion unexposed+1",
16 "OR*prop unexposed", each [prop unexposed] * [Odds ratio], type number),
17      #"Bottom equation portion" = Table.AddColumn(#"Top equation portion", "(OR-
18 1)*(1+pe)", each [#"1+pe"] * [#"OR-1"], type number),
19      #"Odds ratio proportion exposed GIV" = Table.AddColumn(#"Bottom equation
20 portion", "OR prop exposed", each [#"OR*prop unexposed"] / [#"(OR-1)*(1+pe)"],
21 type number),
22      #"Risk ratio proportion exposed GIV" = Table.AddColumn(#"Odds ratio proportion
23 exposed GIV", "RR prop exposed", each [prop unexposed] * [Risk ratio], type
24 number),
25      #"Risk difference proportion exposed GIV" = Table.AddColumn(#"Risk ratio
26 proportion exposed GIV", "RD prop exposed", each [Risk difference] + [prop
27 unexposed], type number),
28      #"Added Conditional Column" = Table.AddColumn(#"Risk difference proportion
29 exposed GIV", "Proportion exposed", each if [Effect measure] = "Risk Ratio" then [RR
30 prop exposed] else if [Effect measure] = "Odds Ratio" then [OR prop exposed] else if
31 [Effect measure] = "Risk Difference" then [RD prop exposed] else null),
32      #"Renamed Columns" = Table.RenameColumns(#"Added Conditional
33 Column",{{"p1", "p1 pre"}, {"p2", "p2 pre"}}),

```

```

1      #"Added Conditional Column1" = Table.AddColumn(#"Renamed Columns", "p1",
2      each if [p1 pre] = null then [Proportion exposed] else [p1 pre]),
3      #"Added Conditional Column2" = Table.AddColumn(#"Added Conditional
4      Column1", "p2", each if [p2 pre] = null then [prop unexposed] else [p2 pre]),
5      #"Adding proportions" = Table.AddColumn(#"Added Conditional Column2", "pb",
6      each [p1] + [p2], type number),
7      #"Divide added proportions by 2" = Table.TransformColumns(#"Adding
8      proportions", {"pb", each _ / 2, type number})),
9      #"1" = Table.AddColumn(#"Divide added proportions by 2", "1", each 1),
10     #"1-pb" = Table.AddColumn(#"1", "qb", each [1] - [pb], type number),
11     #"1-p1" = Table.AddColumn(#"1-pb", "q1", each [1] - [p1], type number),
12     #"1-p2" = Table.AddColumn(#"1-p1", "q2", each [1] - [p2], type number),
13     #"p1-p2" = Table.AddColumn(#"1-p2", "Delta", each [p1] - [p2], type number),
14     #"Absolute only (delta)" = Table.TransformColumns(#"p1-p2", {"Delta",
15     Number.Abs, type number})),
16     #"za (assuming power 0.8)" = Table.AddColumn(#"Absolute only (delta)", "za",
17     each 0.841621234),
18     #"zb (assuming 2 tailed test, alpha 0.05)" = Table.AddColumn(#"za (assuming
19     power 0.8)", "zb", each 1.959963985),
20     #"2" = Table.AddColumn(#"zb (assuming 2 tailed test, alpha 0.05)", "2", each 2),
21     #"2pq" = Table.AddColumn(#"2", "sqrt2pq", each List.Product({[pb], [qb], [2]}), type
22     number),
23     #"sqrt(2pq)" = Table.TransformColumns(#"2pq", {"sqrt2pq", Number.Sqrt, type
24     number})),
25     p1q1 = Table.AddColumn(#"sqrt(2pq)", "p1q1", each [p1] * [q1], type number),
26     p2q2 = Table.AddColumn(p1q1, "p2q2", each [p2] * [q2], type number),
27     sumpqqs = Table.AddColumn(p2q2, "sqrtsumpqqs", each [p1q1] + [p2q2], type
28     number),
29     #"sqrt(sumpqs)" = Table.TransformColumns(sumpqs, {"sqrtsumpqqs",
30     Number.Sqrt, type number})),
31     #"sqrt2pq * za" = Table.AddColumn(#"sqrt(sumpqs)", "sqrt2pq*za", each [sqrt2pq]
32     * [za], type number),
33     #"sqrtsumpqqs * zb" = Table.AddColumn(#"sqrt2pq * za", "sqrtsumpqqs*zb", each
34     [sqrtsumpqqs] * [zb], type number),

```

```

1      #"Adding the two to make the top" = Table.AddColumn("#sqrtsumpqs * zb", "Top",
2  each [#"sqrt2pq*za"] + [#"sqrtsumpqs*zb"], type number),
3      #"Dividing the top by delta" = Table.AddColumn("#Adding the two to make the
4  top", "sss", each [Top] / [Delta], type number),
5      #"Squaring the value" = Table.AddColumn("#Dividing the top by delta", "OIS",
6  each Number.Power([sss], 2), type number),
7      #"Rounding it up to make the OIS" = Table.TransformColumns("#Squaring the
8  value",{{"OIS", each Number.Round(_, 0), type number}}),
9      #"Removed Other Columns1" = Table.SelectColumns("#Rounding it up to make
10 the OIS",{ "Analysis group", "Analysis number", "Analysis name", "Analysis group
11 name", "Outcome", "Intervention grouping", "Experimental intervention", "Control
12 intervention", "Experimental group label", "Control group label", "Effect measure",
13 "Experimental cases", "Experimental N", "Control cases", "CI start", "CI end", "Control
14 N", "OIS"}),
15      #"Inserted Addition" = Table.AddColumn("#Removed Other Columns1", "Total N",
16 each [Experimental N] + [Control N], Int64.Type),
17      #"Inserted Multiplication" = Table.AddColumn("#Inserted Addition", "OIS two
18 arms", each [OIS] * 2, type number),
19      #"OIS proportion calculation" = Table.AddColumn("#Inserted Multiplication", "Prop
20 OIS", each [Total N] / [OIS two arms], type number),
21      #"CI proportion calculation" = Table.AddColumn("#OIS proportion calculation",
22 "Prop CI", each [CI end] / [CI start], type number),
23      #"RR CI proportion conditional" = Table.AddColumn("#CI proportion calculation",
24 "RR Prop CI", each if [Effect measure] = "Risk Ratio" then [Prop CI] else null),
25      #"OR CI proportion conditional" = Table.AddColumn("#RR CI proportion
26 conditional", "OR Prop CI", each if [Effect measure] = "Odds Ratio" then [Prop CI]
27 else null),
28      #"For RR and OR proportions - replace null with 0" = Table.ReplaceValue("#OR CI
29 proportion conditional",null,0,Replacer.ReplaceValue,{"RR Prop CI", "OR Prop CI"}),
30      #"Changed Type1" = Table.TransformColumnTypes("#For RR and OR proportions
31 - replace null with 0",{{"RR Prop CI", type number}, {"OR Prop CI", type number}}),
32      #"Precision conditional determination" = Table.AddColumn("#Changed Type1",
33 "Precision", each if [Effect measure] = "Hazard Ratio" then "Cannot be determined
34 using this method" else if [Effect measure] = "Ratio of Means" then "Cannot be

```

```

1  determined using this method" else if [RR Prop CI] >= 3 then "Very serious concerns
2  regarding optimal information size" else if [OR Prop CI] >= 2.5 then "Very serious
3  concerns regarding optimal information size" else if [Prop OIS] >= 1 then "No
4  concerns regarding optimal information size" else if [Prop OIS] >= 0.3 then "Serious
5  concerns regarding optimal information size" else if [Prop OIS] < 0.3 then "Very
6  serious concerns regarding optimal information size" else "?"),
7      Custom1 = Table.AddColumn("#Precision conditional determination", "OIS
8  Precision", each if [Effect measure] = "Hazard Ratio" then "Not applicable" else if
9  [Effect measure] = "Ratio of Means" then "Not applicable" else if [RR Prop CI] >= 3
10 then "Very serious" else if [OR Prop CI] >= 2.5 then "Very serious" else if [Prop OIS]
11 >= 1 then "Not serious" else if [Prop OIS] >= 0.3 then "Serious" else if [Prop OIS] <
12 0.3 then "Very serious" else "Not serious")
13 in
14     Custom1
15

```

16 **GRADE table**

```

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

```

```

1      #"Splitting by "" at "" to separate follow up time" = Table.SplitColumn(#"Changed
2      Type", "Analysis name", Splitter.SplitTextByDelimiter(" at ", QuoteStyle.Csv),
3      {"Analysis name.1", "Analysis name.2"}),
4      #"Changed Type1" = Table.TransformColumnTypes(#"Splitting by "" at "" to
5      separate follow up time",{{"Analysis name.1", type text}, {"Analysis name.2", type
6      text}}),
7      #"Renamed Columns" = Table.RenameColumns(#"Changed Type1",{{"Analysis
8      name.2", "Follow up time"}}),
9      #"Splitting by "" (" to separate scale information" = Table.SplitColumn(#"Renamed
10     Columns", "Analysis name.1", Splitter.SplitTextByDelimiter(" (", QuoteStyle.Csv),
11     {"Analysis name.1.1", "Analysis name.1.2"}),
12     #"Changed Type2" = Table.TransformColumnTypes(#"Splitting by "" (" to
13     separate scale information",{{"Analysis name.1.1", type text}, {"Analysis name.1.2",
14     type text}}),
15     #"Renamed Columns1" = Table.RenameColumns(#"Changed Type2",{{"Analysis
16     name.1.2", "Scale"}, {"Analysis name.1.1", "Protocol Outcome"}}),
17     #"Removing closing bracket" = Table.ReplaceValue(#"Renamed
18     Columns1",")","",Replacer.ReplaceText,{"Scale"}),
19     #"Making short outcome" = Table.AddColumn(#"Removing closing bracket",
20     "Outcome short", each if [Protocol Outcome] = "Mortality" then "Mort" else if [Protocol
21     Outcome] = "Physical dependency" then "Pdep" else if [Protocol Outcome] = "Falls"
22     then "Falls" else if [Protocol Outcome] = "Readmissions to hospital" then "Read" else
23     if [Protocol Outcome] = "Person/participant generic health-related quality of life" then
24     "HQOL" else if [Protocol Outcome] = "Activities of daily living" then "ADL" else if
25     [Protocol Outcome] = "Extended activities of daily living" then "EADL" else if [Protocol
26     Outcome] = "Length of hospital stay" then "LOS" else if [Protocol Outcome] =
27     "Caregiver strain index" then "CSI" else if [Protocol Outcome] = "Psychological
28     distress/mood" then "PD" else if [Protocol Outcome] = "Carer generic health-related
29     quality of life" then "CHQOL" else if [Protocol Outcome] = "Stroke-specific Patient-
30     Reported Outcome Measures" then "SQOL" else "?"),
31     #"Merged Queries" = Table.NestedJoin(#"Making short outcome", {"Study"},
32     BiasInd, {"Study"}, "Critical appraisal no edits", JoinKind.LeftOuter),

```

```

1      #"Expanded Critical appraisal no edits" = Table.ExpandTableColumn(#"Merged
2      Queries", "Critical appraisal no edits", {"Short population", "Short comparison"},
3      {"Short population", "Short comparison"}),
4      #"Removed Duplicates" = Table.Distinct(#"Expanded Critical appraisal no edits"),
5      #"Removed Columns" = Table.RemoveColumns(#"Removed
6      Duplicates",{"Footnotes", "Study year", "GIV Mean", "GIV SE", "Experimental mean",
7      "Experimental SD", "Experimental cases", "Experimental N", "Control mean", "Control
8      SD", "Control cases", "Control N", "O-E", "Variance", "Weight", "Mean", "CI start", "CI
9      end", "Outcome short", "Short population", "Short comparison"}),
10     #"Duplicated Column" = Table.DuplicateColumn(#"Removed Columns", "Protocol
11     Outcome", "Protocol Outcome - Copy"),
12     #"Pivoted Column" = Table.Pivot(#"Duplicated Column", List.Distinct(#"Duplicated
13     Column"[#"Protocol Outcome - Copy"]), "Protocol Outcome - Copy", "Study",
14     List.Count),
15     #"Inserted Sum" = Table.AddColumn(#"Pivoted Column", "Number of studies",
16     each List.Sum({[Proportion of patients requiring transfusion], [Thrombotic
17     complications], [Duplicate outcome proportion of people requiring transfusion],
18     [Serious adverse events], [#"All-cause mortality"], [Number of units of allogenic blood
19     transfused], [Infection], [Length of stay], [Acute myocardial infarction], [Volume of
20     allogenic blood transfused], [Total blood loss volume]}), type number),
21     #"Inserted Merged Column" = Table.AddColumn(#"Inserted Sum", "GRADE
22     group", each Text.Combine({Text.From([Analysis group], "en-GB"),
23     Text.From([Analysis number], "en-GB")}, "-"), type text),
24     #"Removed Columns1" = Table.RemoveColumns(#"Inserted Merged
25     Column",{"Applicability", "Subgroup", "Proportion of patients requiring transfusion",
26     "Thrombotic complications", "Duplicate outcome proportion of people requiring
27     transfusion", "Serious adverse events", "All-cause mortality", "Number of units of
28     allogenic blood transfused", "Infection", "Length of stay", "Acute myocardial
29     infarction", "Volume of allogenic blood transfused", "Total blood loss volume"}),
30     #"Reordered Columns" = Table.ReorderColumns(#"Removed
31     Columns1",{"Analysis group", "Analysis number", "GRADE group", "Protocol
32     Outcome", "Scale", "Follow up time", "Number of studies"}),
33     #"Merged Queries1" = Table.NestedJoin(#"Reordered Columns", {"GRADE
34     group"}, #"Risk of bias", {"GRADE code"}, "Risk of bias", JoinKind.LeftOuter),

```

```

1      #"Expanded Risk of bias" = Table.ExpandTableColumn(#"Merged Queries1", "Risk
2 of bias", {"RoB", "RoB reason"}, {"RoB", "RoB reason"}),
3      #"Merged Queries3" = Table.NestedJoin(#"Expanded Risk of bias", {"GRADE
4 group"}, #"Inconsistency, units", {"GRADE group"}, "Inconsistency",
5 JoinKind.LeftOuter),
6      #"Expanded Inconsistency" = Table.ExpandTableColumn(#"Merged Queries3",
7 "Inconsistency", {"Importance", "Inconsistency", "Inconsistency reason", "Other
8 considerations", "ROBINS-I used?", "Study type", "Units"}, {"Importance",
9 "Inconsistency.1", "Inconsistency reason", "Other considerations", "ROBINS-I used?",
10 "Study type", "Units"}),
11      #"Renamed Columns2" = Table.RenameColumns(#"Expanded
12 Inconsistency", {"Inconsistency.1", "Inconsistency"}),
13      #"Merged Queries2" = Table.NestedJoin(#"Renamed Columns2", {"GRADE
14 group"}, Indirectness, {"GRADE code"}, "Indirectness", JoinKind.LeftOuter),
15      #"Expanded Indirectness" = Table.ExpandTableColumn(#"Merged Queries2",
16 "Indirectness", {"Indirectness", "Indirectness reason"}, {"Indirectness", "Indirectness
17 reason"}),
18      #"Merged Queries4" = Table.NestedJoin(#"Expanded Indirectness", {"GRADE
19 group"}, Imprecision, {"GRADE group"}, "Imprecision", JoinKind.LeftOuter),
20      #"Expanded Imprecision" = Table.ExpandTableColumn(#"Merged Queries4",
21 "Imprecision", {"Imprecision", "Imprecision reason", "Sample size"}, {"Imprecision",
22 "Imprecision reason", "Sample size"}),
23      #"Merged Queries5" = Table.NestedJoin(#"Expanded Imprecision", {"GRADE
24 group"}, Timepoint, {"GRADE code"}, "Timepoint", JoinKind.LeftOuter),
25      #"Expanded Timepoint" = Table.ExpandTableColumn(#"Merged Queries5",
26 "Timepoint", {"Average timepoint (months)"}, {"Average timepoint (months)}),
27      #"Added Conditional Column1" = Table.AddColumn(#"Expanded Timepoint",
28 "ROBINS-I adjustment", each if [Study type] = "Randomised trials" then 0 else if
29 [#"ROBINS-I used?"] = true then 0 else 2),
30      #"Calculating RoB relative number" = Table.AddColumn(#"Added Conditional
31 Column1", "RoB number", each if [RoB] = "Not serious" then 0 else if [RoB] =
32 "Serious" then 1 else if [RoB] = "Very serious" then 2 else "?"),
33      #"Calculating indirectness relative number" = Table.AddColumn(#"Calculating RoB
34 relative number", "Indirectness number", each if [Indirectness] = "Not serious" then 0

```

```

1  else if [Indirectness] = "Serious" then 1 else if [Indirectness] = "Very serious" then 2
2  else "?"),
3      #"Calculating inconsistency number" = Table.AddColumn(#"Calculating
4  indirectness relative number", "Inconsistency number", each if [Inconsistency] = "Not
5  serious" then 0 else if [Inconsistency] = "Serious" then 1 else if [Inconsistency] =
6  "Very serious" then 2 else "?"),
7      #"Calculating imprecision relative number" = Table.AddColumn(#"Calculating
8  inconsistency number", "Imprecision number", each if [Imprecision] = "Not serious"
9  then 0 else if [Imprecision] = "Serious" then 1 else if [Imprecision] = "Very serious"
10 then 2 else "?"),
11      #"Changed Type3" = Table.TransformColumnTypes(#"Calculating imprecision
12 relative number",{{"RoB number", Int64.Type}, {"Indirectness number", Int64.Type},
13 {"Inconsistency number", Int64.Type}, {"Imprecision number", Int64.Type},
14 {"ROBINS-I adjustment", Int64.Type}}),
15      #"Inserted Sum1" = Table.AddColumn(#"Changed Type3", "GRADE number",
16 each List.Sum({{"ROBINS-I adjustment"}, [RoB number], [Indirectness number],
17 [Inconsistency number], [Imprecision number]}), Int64.Type),
18      #"Added Conditional Column" = Table.AddColumn(#"Inserted Sum1", "GRADE",
19 each if [GRADE number] = 0 then "High" else if [GRADE number] = 1 then
20 "Moderate" else if [GRADE number] = 2 then "Low" else if [GRADE number] >= 3
21 then "Very low" else "?"),
22      #"Inserted Merged Column1" = Table.AddColumn(#"Added Conditional Column",
23 "Full reasons", each Text.Combine({[RoB reason], [Inconsistency reason],
24 [Indirectness reason], [Imprecision reason]}), "
25 "), type text),
26      #"Completing reasons" = Table.AddColumn(#"Inserted Merged Column1",
27 "Reasons", each if [Full reasons] = "" then "No downgrading required" else [Full
28 reasons]),
29      #"Removed Columns5" = Table.RemoveColumns(#"Completing reasons",{"Full
30 reasons"}),
31      #"Analyst edits for scale column" = Table.AddColumn(#"Removed Columns5",
32 "Outcome scale", each if [Scale] = "days" then "Not applicable" else if [Scale] = "mL"
33 then "Not applicable" else if [Scale] = "units" then "Not applicable" else if [Scale] <>

```



```

1 null then [Scale] else if [Protocol Outcome] = "Mortality" then "Not applicable" else
2 "Not applicable"),
3   #"Risk of bias short reason" = Table.AddColumn(#"Analyst edits for scale column",
4 "Short RoB reason", each if [RoB reason] <> null then "Risk of Bias" else null),
5   #"Inconsistency short reason" = Table.AddColumn(#"Risk of bias short reason",
6 "Inconsistency short reason", each if [Inconsistency reason] <> null then
7 "Inconsistency" else null),
8   #"Indirectness short reason" = Table.AddColumn(#"Inconsistency short reason",
9 "Indirectness short reason", each if [Indirectness reason] <> null then "Indirectness"
10 else null),
11   #"Imprecision short reason" = Table.AddColumn(#"Indirectness short reason",
12 "Imprecision short reason", each if [Imprecision reason] <> null then "Imprecision"
13 else null),
14   #"Combining short reasons" = Table.AddColumn(#"Imprecision short reason",
15 "Short reason", each Text.Combine({[Short RoB reason], [Inconsistency short
16 reason], [Indirectness short reason], [Imprecision short reason]}, ", "), type text),
17   #"Removed Columns2" = Table.RemoveColumns(#"Combining short
18 reasons",{ "Short RoB reason", "Inconsistency short reason", "Indirectness short
19 reason", "Imprecision short reason"}),
20   #"Added Custom" = Table.AddColumn(#"Removed Columns2", "Scale: ", each
21 "Scale: "),
22   #"Added Custom1" = Table.AddColumn(#"Added Custom", "Units: ", each "Units:
23 "),
24   #"Added Custom2" = Table.AddColumn(#"Added Custom1", "Study types:", each
25 "Study types: "),
26   #"Inserted Merged Column2" = Table.AddColumn(#"Added Custom2", "Scale:
27 value", each Text.Combine({[#"Scale: "], [Outcome scale]}, ""), type text),
28   #"Inserted Merged Column3" = Table.AddColumn(#"Inserted Merged Column2",
29 "Units: value", each Text.Combine({[#"Units: "], [Units]}, ""), type text),
30   #"Inserted Merged Column5" = Table.AddColumn(#"Inserted Merged Column3",
31 "Study type: value", each Text.Combine({[#"Study types:"], [Study type]}, ""), type
32 text),
33   #"Removed Columns3" = Table.RemoveColumns(#"Inserted Merged
34 Column5",{ "Scale: ", "Units: ", "Study types:"}),

```

```

1      # "Inserted Merged Column4" = Table.AddColumn(#"Removed Columns3",
2      "Outcome table", each Text.Combine({[Protocol Outcome], [Follow up time]}, " at "),
3      type text),
4      # "Lowercased Text" = Table.TransformColumns(#"Inserted Merged
5      Column4",{{"Scale: value", Text.Lower, type text}, {"Units: value", Text.Lower, type
6      text}, {"Study type: value", Text.Lower, type text}}),
7      # "Replaced Value" = Table.ReplaceValue(#"Lowercased
8      Text","ml","mL",Replacer.ReplaceText,{"Units: value"}),
9      # "Merged Columns" = Table.CombineColumns(#"Replaced Value",{ "Outcome
10     table", "Study type: value", "Scale: value", "Units:
11     value"},Combiner.CombineTextByDelimiter(", ", QuoteStyle.None),"Outcome full
12     table"),
13     # "Added Custom3" = Table.AddColumn(#"Merged Columns", "Risk of bias: ", each
14     "Risk of bias: "),
15     # "Added Custom4" = Table.AddColumn(#"Added Custom3", "Indirectness: ", each
16     "Indirectness: "),
17     # "Added Custom5" = Table.AddColumn(#"Added Custom4", "Inconsistency: ",
18     each "Inconsistency: "),
19     # "Added Custom6" = Table.AddColumn(#"Added Custom5", "Imprecision: ", each
20     "Imprecision: "),
21     # "Added Custom7" = Table.AddColumn(#"Added Custom6", "Other concerns: ",
22     each "Other considerations: "),
23     # "Inserted Merged Column6" = Table.AddColumn(#"Added Custom7", "Risk of
24     bias: value", each Text.Combine({[#"Risk of bias: "], [RoB]}, "" ), type text),
25     # "Inserted Merged Column7" = Table.AddColumn(#"Inserted Merged Column6",
26     "Indirectness: value", each Text.Combine({[#"Indirectness: "], [Indirectness]}, "" ), type
27     text),
28     # "Inserted Merged Column8" = Table.AddColumn(#"Inserted Merged Column7",
29     "Inconsistency: value", each Text.Combine({[#"Inconsistency: "], [Inconsistency]}, "" ),
30     type text),
31     # "Inserted Merged Column9" = Table.AddColumn(#"Inserted Merged Column8",
32     "Imprecision: value", each Text.Combine({[#"Imprecision: "], [Imprecision]}, "" ), type
33     text),

```

```

1      #"Inserted Merged Column10" = Table.AddColumn(#"Inserted Merged Column9",
2      "Other considerations: value", each Text.Combine({#"Other concerns: ", [Other
3      considerations]}, ""), type text),
4      #"Inserted Merged Column11" = Table.AddColumn(#"Inserted Merged Column10",
5      "GRADE components", each Text.Combine({#"Risk of bias: value", [#"Indirectness:
6      value"], [#"Inconsistency: value"], [#"Imprecision: value"], [#"Other considerations:
7      value"]}, "
8      "), type text),
9      #"Removed Columns4" = Table.RemoveColumns(#"Inserted Merged
10     Column11",{"Risk of bias: ", "Indirectness: ", "Inconsistency: ", "Imprecision: ", "Other
11     concerns: ", "Risk of bias: value", "Indirectness: value", "Inconsistency: value",
12     "Imprecision: value", "Other considerations: value"})
13     in
14     #"Removed Columns4"
15

```

16 **Appendix B Review [B] Power M query for GRADE** 17 **automation**

18 **Data rows**

```

19     let
20         Source = Csv.Document(Web.Contents("source"),[Delimiter=","]),
21         #"Promoted Headers" = Table.PromoteHeaders(Source,
22         [PromoteAllScalars=true]),
23         #"Changed Type" = Table.TransformColumnTypes(#"Promoted
24         Headers",{{"Analysis group", Int64.Type}, {"Analysis number", Int64.Type}, {"Analysis
25         name", type text}, {"Subgroup", type any}, {"Applicability", type text}, {"Study", type
26         text}, {"Study year", Int64.Type}, {"GIV Mean", type number}, {"GIV SE", type
27         number}, {"Experimental mean", type number}, {"Experimental SD", type number},
28         {"Experimental cases", Int64.Type}, {"Experimental N", Int64.Type}, {"Control mean",
29         type number}, {"Control SD", type number}, {"Control cases", Int64.Type}, {"Control
30         N", Int64.Type}, {"O-E", type any}, {"Variance", type any}, {"Weight", type number},
31         {"Mean", type number}, {"CI start", type number}, {"CI end", type number},
32         {"Footnotes", type any}}),

```

```

1      #"Splitting by "" at "" to separate follow up time" = Table.SplitColumn(#"Changed
2      Type", "Analysis name", Splitter.SplitTextByDelimiter(" at ", QuoteStyle.Csv),
3      {"Analysis name.1", "Analysis name.2"}),
4      #"Changed Type1" = Table.TransformColumnTypes(#"Splitting by "" at "" to
5      separate follow up time",{{"Analysis name.1", type text}, {"Analysis name.2", type
6      text}}),
7      #"Renamed Columns" = Table.RenameColumns(#"Changed Type1",{{"Analysis
8      name.2", "Follow up time"}}),
9      #"Splitting by "" (" to separate scale information" = Table.SplitColumn(#"Renamed
10     Columns", "Analysis name.1", Splitter.SplitTextByDelimiter(" (", QuoteStyle.Csv),
11     {"Analysis name.1.1", "Analysis name.1.2"}),
12     #"Changed Type2" = Table.TransformColumnTypes(#"Splitting by "" (" to
13     separate scale information",{{"Analysis name.1.1", type text}, {"Analysis name.1.2",
14     type text}}),
15     #"Renamed Columns1" = Table.RenameColumns(#"Changed Type2",{{"Analysis
16     name.1.2", "Scale"}, {"Analysis name.1.1", "Protocol Outcome"}}),
17     #"Removing closing bracket" = Table.ReplaceValue(#"Renamed
18     Columns1",")","",Replacer.ReplaceText,{"Scale"}),
19     #"Making short outcome" = Table.AddColumn(#"Removing closing bracket",
20     "Outcome short", each if [Protocol Outcome] = "Thromboembolic events after
21     surgery" then "Thromboembolic events" else if [Protocol Outcome] = "All-cause
22     mortality" then "Mortality" else if [Protocol Outcome] = "Pulmonary embolism" then
23     "PE" else if [Protocol Outcome] = "Deep vein thrombosis" then "DVT" else if [Protocol
24     Outcome] = "Myocardial infarction" then "MI" else if [Protocol Outcome] = "Ischaemic
25     stroke" then "Stroke" else if [Protocol Outcome] = "Infection" then "Infection" else if
26     [Protocol Outcome] = "All-cause readmission" then "Readmission" else if [Protocol
27     Outcome] = "Seizures" then "Seizures" else if [Protocol Outcome] = "Reoperation"
28     then "Reoperation" else "?"),
29     #"Merged Queries" = Table.NestedJoin(#"Making short outcome", {"Study ID"},
30     #"Study level data", {"Study ID"}, "BiasInd", JoinKind.LeftOuter),
31     #"Expanded Critical appraisal no edits" = Table.ExpandTableColumn(#"Merged
32     Queries", "BiasInd", {"Short population", "Short comparison"}, {"Short population",
33     "Short comparison"}),
34     #"Removed Duplicates" = Table.Distinct(#"Expanded Critical appraisal no edits"),

```

```

1      #"Removed Columns" = Table.RemoveColumns(#"Removed
2 Duplicates",{ "Footnotes"}),
3      #"Inserted Merged Column" = Table.AddColumn(#"Removed Columns", "GRADE
4 number", each Text.Combine({Text.From([Analysis group], "en-GB"),
5 Text.From([Analysis number], "en-GB")}, "-"), type text),
6      #"Filtered Rows" = Table.SelectRows(#"Inserted Merged Column", each ([GRADE
7 number] <> "5-1" and [GRADE number] <> "5-10" and [GRADE number] <> "5-11"
8 and [GRADE number] <> "5-12" and [GRADE number] <> "5-13" and [GRADE
9 number] <> "5-14" and [GRADE number] <> "5-2" and [GRADE number] <> "5-3"
10 and [GRADE number] <> "5-4" and [GRADE number] <> "5-5" and [GRADE number]
11 <> "5-6" and [GRADE number] <> "5-7" and [GRADE number] <> "5-8" and [GRADE
12 number] <> "5-9" and [GRADE number] <> "6-1" and [GRADE number] <> "6-10"
13 and [GRADE number] <> "6-11" and [GRADE number] <> "6-12" and [GRADE
14 number] <> "6-13" and [GRADE number] <> "6-14" and [GRADE number] <> "6-15"
15 and [GRADE number] <> "6-16" and [GRADE number] <> "6-17" and [GRADE
16 number] <> "6-18" and [GRADE number] <> "6-19" and [GRADE number] <> "6-2"
17 and [GRADE number] <> "6-20" and [GRADE number] <> "6-21" and [GRADE
18 number] <> "6-3" and [GRADE number] <> "6-4" and [GRADE number] <> "6-5" and
19 [GRADE number] <> "6-6" and [GRADE number] <> "6-7" and [GRADE number] <>
20 "6-8" and [GRADE number] <> "6-9")),
21      #"Replaced Value" = Table.ReplaceValue(#"Filtered
22 Rows",null,0,Replacer.ReplaceValue,{"Mean", "CI start", "CI end"}),
23      #"Replaced Value1" = Table.ReplaceValue(#"Replaced
24 Value",0,0.0001,Replacer.ReplaceValue,{"CI end"}),
25      #"Risk intervention" = Table.AddColumn(#"Replaced Value1", "Risk Intervention",
26 each [Experimental cases] / [Experimental N], type number),
27      #"Risk control" = Table.AddColumn(#"Risk intervention", "Risk control", each
28 [Control cases] / [Control N], type number),
29      #"Risk ratio" = Table.AddColumn(#"Risk control", "Risk ratio", each [Risk
30 Intervention] / [Risk control], type number),
31      #"1/a" = Table.AddColumn(#"Risk ratio", "1/a", each Number.Power([Experimental
32 cases], -1), Int64.Type),
33      #"1/c" = Table.AddColumn(#"1/a", "1/c", each Number.Power([Control cases], -1),
34 Int64.Type),

```

```

1      #"a+b" = Table.AddColumn("#1/c", "a+b", each [Experimental cases] +
2      [Experimental N], Int64.Type),
3      #"c+d" = Table.AddColumn("#a+b", "c+d", each [Control cases] + [Control N],
4      Int64.Type),
5      #"1/a+b" = Table.AddColumn("#c+d", "1/a+b", each Number.Power([#"a+b"], -1),
6      type number),
7      #"1/c+d" = Table.AddColumn("#1/a+b", "1/c+d", each Number.Power([#"c+d"], -1),
8      type number),
9      #"Inserted Natural Logarithm" = Table.AddColumn("#1/c+d", "ln(RR)", each
10     Number.Ln([Risk ratio]), type number),
11     #"Inserted Addition" = Table.AddColumn("#Inserted Natural Logarithm", "1/a +
12     1/c", each [#"1/a"] + [#"1/c"], Int64.Type),
13     #"Inserted Subtraction" = Table.AddColumn("#Inserted Addition", "Subtraction",
14     each [#"1/a + 1/c"] - [#"1/a+b"], type number),
15     #"Inserted Subtraction1" = Table.AddColumn("#Inserted Subtraction",
16     "Subtraction.1", each [Subtraction] - [#"1/c+d"], type number),
17     #"Inserted Square Root" = Table.AddColumn("#Inserted Subtraction1", "Square
18     Root", each Number.Sqrt([Subtraction.1]), type number),
19     #"1.96*SQRT" = Table.AddColumn("#Inserted Square Root", "1.96*SQRT", each
20     [Square Root] * 1.96, type number),
21     #"ln(RR)+1.96*SQRT" = Table.AddColumn("#1.96*SQRT", "Addition", each
22     [#"ln(RR)"] + [#"1.96*SQRT"], type number),
23     #"ln(RR)-1.96*SQRT" = Table.AddColumn("#ln(RR)+1.96*SQRT", "Subtraction.2",
24     each [#"ln(RR)"] - [#"1.96*SQRT"], type number),
25     #"97.5% CI" = Table.AddColumn("#ln(RR)-1.96*SQRT", "97.5% CI", each
26     Number.Exp([Addition]), type number),
27     #"2.5% CI" = Table.AddColumn("#97.5% CI", "2.5% CI", each
28     Number.Exp([Subtraction.2]), type number),
29     #"Removed Columns1" = Table.RemoveColumns("#2.5% CI",{"1/a", "1/c", "a+b",
30     "c+d", "1/a+b", "1/c+d", "ln(RR)", "1/a + 1/c", "Subtraction", "Subtraction.1", "Square
31     Root", "1.96*SQRT", "Addition", "Subtraction.2", "Risk Intervention", "Risk control"}),
32     #"Added Conditional Column" = Table.AddColumn("#Removed Columns1", "Risk
33     ratio f", each if [Risk ratio] = null then null else if [Risk ratio] >= 0.0001 then [Risk
34     ratio] else null),

```

```

1      #"Added Conditional Column1" = Table.AddColumn(#"Added Conditional Column",
2      "2.5% CI f", each if [#"2.5% CI"] = null then null else if [#"2.5% CI"] >= 1E-05 then
3      [#"2.5% CI"] else null),
4      #"Added Conditional Column2" = Table.AddColumn(#"Added Conditional
5      Column1", "97.5% CI f", each if [#"97.5% CI"] = null then null else if [#"97.5% CI"] >=
6      1E-05 then [#"97.5% CI"] else null),
7      #"Filtered Rows1" = Table.SelectRows(#"Added Conditional Column2", each
8      ([Scale] <> "risk difference")),
9      #"Replaced Value2" = Table.ReplaceValue(#"Filtered Rows1", "risk
10     ratio", null, Replacer.ReplaceValue, {"Scale"}),
11     #"Inserted Merged Column1" = Table.AddColumn(#"Replaced Value2", "Study ID-
12     Outcome", each Text.Combine({[Study ID], [Protocol Outcome], [Scale]}, "-"), type
13     text)
14     in
15     #"Inserted Merged Column1"

```

16 Overall estimates

```

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

```

```

1  {"Experimental N", Int64.Type}, {"Control cases", Int64.Type}, {"Control N",
2  Int64.Type}, {"Mean", type number}, {"CI start", type number}, {"CI end", type
3  number}, {"PI start", type any}, {"PI end", type any}, {"Heterogeneity Tau²", type any},
4  {"Tau² CI start", type any}, {"Tau² CI end", type any}, {"Heterogeneity Chi²", type
5  number}, {"Heterogeneity df", Int64.Type}, {"Heterogeneity P", type number},
6  {"Heterogeneity I²", type number}, {"Effect Z", type number}, {"Effect P", type
7  number}, {"Subgroup Chi²", type any}, {"Subgroup df", type any}, {"Subgroup P", type
8  any}, {"Subgroup I²", type any}, {"ID", type text})),
9    # "Removed Columns" = Table.RemoveColumns(# "Changed Type", {"PI start", "PI
10   end", "Heterogeneity Tau²", "Tau² CI start", "Tau² CI end", "Heterogeneity Chi²",
11   "Heterogeneity df", "Heterogeneity P", "Effect Z", "Effect P", "Subgroup Chi²",
12   "Subgroup df", "Subgroup P", "ID", "CI method", "Test for subgroup differences",
13   "Prediction interval", "Swap event and non-event", "Tau² CI", "Heterogeneity
14   estimator", "Experimental intervention", "Control intervention"}),
15    # "Inserted Merged Column" = Table.AddColumn(# "Removed Columns", "GRADE
16   group", each Text.Combine({Text.From([Analysis group], "en-GB"),
17   Text.From([Analysis number], "en-GB")}, "-"), type text),
18    # "Analyst editable control risk rate (proportion)" = Table.AddColumn(# "Inserted
19   Merged Column", "Control risk rate (proportion)", each if [Outcome] = "a" then 0.2
20   else null),
21    # "Changed Type6" = Table.TransformColumnTypes(# "Analyst editable control risk
22   rate (proportion)", {"Control risk rate (proportion)", type number}),
23    # "Adjusted for editable control n calculation" = Table.AddColumn(# "Changed
24   Type6", "Adjusted control N", each [# "Control risk rate (proportion)"] * [Control N],
25   type number),
26    # "Changed Type8" = Table.TransformColumnTypes(# "Adjusted for editable control
27   n calculation", {"Adjusted control N", Int64.Type}),
28    # "Merged Queries" = Table.NestedJoin(# "Changed Type8", {"GRADE group"},
29   # "Median baseline rate", {"GRADE number"}, "Median baseline rate",
30   JoinKind.LeftOuter),
31    # "Expanded Median baseline rate" = Table.ExpandTableColumn(# "Merged
32   Queries", "Median baseline rate", {"Median control group rate"}, {"Median control
33   group rate"}),

```



```

1      #"Adjusted for median control group rate" = Table.AddColumn(#"Expanded
2 Median baseline rate", "Median control n", each [Median control group rate] * [Control
3 N], type number),
4      #"Changed Type7" = Table.TransformColumnTypes(#"Adjusted for median control
5 group rate",{{"Median control n", Int64.Type}}),
6      #"Renamed Columns4" = Table.RenameColumns(#"Changed Type7",{{"Control
7 N", "True Control N"}}),
8      #"Analyst editable choice of control group N" = Table.AddColumn(#"Renamed
9 Columns4", "Control N", each if [Outcome] = "a" then [Adjusted control N] else if
10 [Outcome] = "b" then [Median control n] else if [Outcome] = "c" then [True Control N]
11 else [True Control N]),
12      #"Calculating non-events 1" = Table.AddColumn(#"Analyst editable choice of
13 control group N", "Non-events 1", each [Experimental N] - [Experimental cases],
14 Int64.Type),
15      #"Calculating non-events 2" = Table.AddColumn(#"Calculating non-events 1",
16 "Non-events 2", each [Control N] - [Control cases], Int64.Type),
17      #"Calculating control rate" = Table.AddColumn(#"Calculating non-events 2",
18 "Baseline control rate", each ([Control cases] / [Control N]) * 1000, Int64.Type),
19      #"Inserted Multiplication6" = Table.AddColumn(#"Calculating control rate",
20 "Baseline control rate/10,000", each [Baseline control rate] * 10, type number),
21      #"Changed Type9" = Table.TransformColumnTypes(#"Inserted
22 Multiplication6",{{"Baseline control rate", Int64.Type}, {"Baseline control rate/10,000",
23 Int64.Type}}),
24      #"per 1000 people" = Table.AddColumn(#"Changed Type9", "per 1000 people",
25 each " per 10,000 people"),
26      #"Making control group rate" =
27 Table.CombineColumns(Table.TransformColumnTypes(#"per 1000 people",
28 {"Baseline control rate/10,000", type text}, "en-GB"),{"Baseline control rate/10,000",
29 "per 1000 people"},Combiner.CombineTextByDelimiter("'", QuoteStyle.None),"Control
30 group rate"),
31      #"Replaced Value" = Table.ReplaceValue(#"Making control group rate"," per
32 10,000 people","Not applicable",Replacer.ReplaceValue,{"Control group rate"}),
33      #"Calculating risk1" = Table.AddColumn(#"Replaced Value", "Risk1", each
34 [Experimental cases] / [Experimental N], type number),

```

```

1      #"Calculating risk2" = Table.AddColumn(#"Calculating risk1", "Risk2", each
2      [Control cases] / [Control N], type number),
3      #"Renamed Columns" = Table.RenameColumns(#"Calculating
4      risk2",{{"Experimental cases", "Number of events 1"}, {"Experimental N", "n1"},
5      {"Control cases", "Number of events 2"}, {"Control N", "n2"}}),
6      #"Calculating risk difference" = Table.AddColumn(#"Renamed Columns", "Risk
7      difference", each [Risk1] - [Risk2], type number),
8      #"Inserted Multiplication1" = Table.AddColumn(#"Calculating risk difference",
9      "Event1xNonevent1", each [Number of events 1] * [#"Non-events 1"], Int64.Type),
10     #"Inserted Multiplication2" = Table.AddColumn(#"Inserted Multiplication1",
11     "EventxNonevent2", each [Number of events 2] * [#"Non-events 2"], Int64.Type),
12     #"Inserted Cube" = Table.AddColumn(#"Inserted Multiplication2", "n1^3", each
13     Number.Power([#"Non-events 1"], 3), Int64.Type),
14     #"Inserted Cube1" = Table.AddColumn(#"Inserted Cube", "n2^3", each
15     Number.Power([n2], 3), Int64.Type),
16     #"Inserted Division3" = Table.AddColumn(#"Inserted Cube1",
17     "eventxnonevent1/n1^3", each [Event1xNonevent1] / [#"n1^3"], type number),
18     #"Inserted Division4" = Table.AddColumn(#"Inserted Division3", "Division", each
19     [EventxNonevent2] / [#"n2^3"], type number),
20     #"Renamed Columns1" = Table.RenameColumns(#"Inserted
21     Division4",{{"Division", "eventxnonevent2/n2^3"}}),
22     #"Inserted Addition2" = Table.AddColumn(#"Renamed Columns1",
23     "eventxnonevent1/n1^3 + eventxnonevent2/n2^3", each [#"eventxnonevent1/n1^3"] +
24     [#"eventxnonevent2/n2^3"], type number),
25     #"Inserted Square Root1" = Table.AddColumn(#"Inserted Addition2", "SE(RD)",
26     each Number.Sqrt([#"eventxnonevent1/n1^3 + eventxnonevent2/n2^3"]), type
27     number),
28     #"Inserted Multiplication3" = Table.AddColumn(#"Inserted Square Root1",
29     "1.96*SE", each [#"SE(RD)"] * 1.96, type number),
30     #"Inserted Subtraction4" = Table.AddColumn(#"Inserted Multiplication3", "RD 2.5%
31     CI", each [Risk difference] - [#"1.96*SE"], type number),
32     #"Inserted Addition3" = Table.AddColumn(#"Inserted Subtraction4", "RD 97.5%
33     CI", each [Risk difference] + [#"1.96*SE"], type number),

```

```

1      #"Removed Columns3" = Table.RemoveColumns(#"Inserted
2      Addition3",{ "Event1xNonevent1", "EventxNonevent2", "n1^3", "n2^3",
3      "eventxnonevent1/n1^3", "eventxnonevent2/n2^3", "eventxnonevent1/n1^3 +
4      eventxnonevent2/n2^3", "SE(RD)", "1.96*SE", "Non-events 1", "Non-events 2",
5      "Risk1"}),
6      #"Calculating events/1000 people" = Table.AddColumn(#"Removed Columns3",
7      "Events/1000 RD", each [Risk difference] * 1000, type number),
8      #"Calculating 2.5% CI" = Table.AddColumn(#"Calculating events/1000 people",
9      "Events/1000 2.5% CI RD", each [#"RD 2.5% CI"] * 1000, type number),
10     #"Calculating 97.5% CI" = Table.AddColumn(#"Calculating 2.5% CI", "Events/1000
11     97.5% CI RD", each [#"RD 97.5% CI"] * 1000, type number),
12     #"Calculating events/1000 people mean" = Table.AddColumn(#"Calculating 97.5%
13     CI", "Events/1000 mean", each [Mean] * 1000, type number),
14     #"Calculating 2.5% CI mean" = Table.AddColumn(#"Calculating events/1000
15     people mean", "Events/1000 2.5% CI mean", each [#"CI start"] * 1000, type number),
16     #"Calculating 97.5% CI mean" = Table.AddColumn(#"Calculating 2.5% CI mean",
17     "Events/1000 97.5% CI mean", each [#"CI end"] * 1000, type number),
18     #"Calculating events/10000 people" = Table.AddColumn(#"Calculating 97.5% CI
19     mean", "Events/10000 RD", each [Risk difference] * 10000, type number),
20     #"Calculating 2.5% CI 10000" = Table.AddColumn(#"Calculating events/10000
21     people", "Events/10000 2.5% CI RD", each [#"RD 2.5% CI"] * 10000, type number),
22     #"Calculating 97.5% CI 10000" = Table.AddColumn(#"Calculating 2.5% CI 10000",
23     "Events/10000 97.5% CI RD", each [#"RD 97.5% CI"] * 10000, type number),
24     #"Calculating events/10000 people mean" = Table.AddColumn(#"Calculating
25     97.5% CI 10000", "Events/10000 mean", each [Mean] * 10000, type number),
26     #"Calculating 2.5% CI mean 10000" = Table.AddColumn(#"Calculating
27     events/10000 people mean", "Events/10000 2.5% CI mean", each [#"CI start"] *
28     10000, type number),
29     #"Calculating 97.5% CI mean 10000" = Table.AddColumn(#"Calculating 2.5% CI
30     mean 10000", "Events/10000 97.5% CI mean", each [#"CI end"] * 10000, type
31     number),
32     #"Changed Type1" = Table.TransformColumnTypes(#"Calculating 97.5% CI mean
33     10000",{{"Events/1000 97.5% CI RD", Int64.Type}, {"Events/1000 2.5% CI RD",
34     Int64.Type}, {"Events/1000 RD", Int64.Type}, {"Events/1000 97.5% CI mean",

```

```

1  Int64.Type}, {"Events/1000 2.5% CI mean", Int64.Type}, {"Events/1000 mean",
2  Int64.Type}, {"Events/10000 RD", Int64.Type}, {"Events/10000 2.5% CI RD",
3  Int64.Type}, {"Events/10000 97.5% CI RD", Int64.Type}, {"Events/10000 mean",
4  Int64.Type}, {"Events/10000 2.5% CI mean", Int64.Type}, {"Events/10000 97.5% CI
5  mean", Int64.Type}}),
6  #"Analyst choice events/1000 method" = Table.AddColumn(#"Changed Type1",
7  "Events/1000", each if [Effect measure] = "Risk Difference" then [#"Events/1000
8  mean"] else [#"Events/1000 RD"]),
9  #"Analyst choice 2.5% CI" = Table.AddColumn(#"Analyst choice events/1000
10 method", "Events/1000 2.5% CI", each if [Effect measure] = "Risk Difference" then
11 [#"Events/1000 2.5% CI mean"] else [#"Events/1000 2.5% CI RD"]),
12  #"Analyst choice 97.5% CI" = Table.AddColumn(#"Analyst choice 2.5% CI",
13 "Events/1000 97.5% CI", each if [Effect measure] = "Risk Difference" then
14 [#"Events/1000 97.5% CI mean"] else [#"Events/1000 97.5% CI RD"]),
15  #"Analyst choice events/10000 method" = Table.AddColumn(#"Analyst choice
16 97.5% CI", "Events/10000", each if [Effect measure] = "Risk Difference" then
17 [#"Events/10000 mean"] else [#"Events/10000 RD"]),
18  #"Analyst choice 2.5% CI 10000" = Table.AddColumn(#"Analyst choice
19 events/10000 method", "Events/10000 2.5% CI", each if [Effect measure] = "Risk
20 Difference" then [#"Events/10000 2.5% CI mean"] else [#"Events/10000 2.5% CI
21 RD"]),
22  #"Analyst choice 97.5% CI 10000" = Table.AddColumn(#"Analyst choice 2.5% CI
23 10000", "Events/10000 97.5% CI", each if [Effect measure] = "Risk Difference" then
24 [#"Events/10000 97.5% CI mean"] else [#"Events/10000 97.5% CI RD"]),
25  #"Extracting RD direction of effect" = Table.AddColumn(#"Analyst choice 97.5% CI
26 10000", "Sign RD", each Number.Sign([#"Events/1000"]), Int64.Type),
27  #"Extracting 2.5% CI direction of effect" = Table.AddColumn(#"Extracting RD
28 direction of effect", "Sign RD 2.5%", each Number.Sign([#"Events/1000 2.5% CI"]),
29 Int64.Type),
30  #"Extracting 97.5% CI direction of effect" = Table.AddColumn(#"Extracting 2.5% CI
31 direction of effect", "Sign RD 97.5%", each Number.Sign([#"Events/1000 97.5% CI"]),
32 Int64.Type),
33  #"Determining words after RD" = Table.AddColumn(#"Extracting 97.5% CI
34 direction of effect", "RD wording", each if [Sign RD] = 1 then " more events per

```

```

1  10,000 people, " else if [Sign RD] = -1 then " fewer events per 10,000 people, " else if
2  [Sign RD] = 0 then " fewer events per 10,000 people, " else if [Sign RD] = null then
3  null else "?"),
4      #"Determining words after RD 2.5% CI" = Table.AddColumn(#"Determining words
5  after RD", "RD 2.5% CI wording", each if [#"Sign RD 2.5%"] = 1 then " more to " else
6  if [#"Sign RD 2.5%"] = -1 then " fewer to " else if [#"Sign RD 2.5%"] = 0 then " fewer
7  to " else if [#"RD 2.5% CI"] = null then null else "?"),
8      #"Determining words after RD 97.5% CI" = Table.AddColumn(#"Determining
9  words after RD 2.5% CI", "RD 97.5% CI wording", each if [#"Sign RD 97.5%"] = 1
10 then " more" else if [#"Sign RD 97.5%"] = -1 then " fewer" else if [#"Sign RD 97.5%"]
11 = 0 then " fewer" else if [#"Sign RD 97.5%"] = null then null else "?"),
12      #"Maintain RD direction" = Table.DuplicateColumn(#"Determining words after RD
13 97.5% CI", "Events/1000", "Events/1000 direction"),
14      #"Maintain RD 2.5% direction" = Table.DuplicateColumn(#"Maintain RD direction",
15  "Events/1000 2.5% CI", "Events/1000 2.5% CI direction"),
16      #"Maintain RD 97.5% direction" = Table.DuplicateColumn(#"Maintain RD 2.5%
17  direction", "Events/1000 97.5% CI", "Events/1000 97.5% CI direction"),
18      #"Calculated Absolute Value" = Table.TransformColumns(#"Maintain RD 97.5%
19  direction",{{"Events/1000", Number.Abs, Int64.Type}, {"Events/1000 2.5% CI",
20  Number.Abs, Int64.Type}, {"Events/1000 97.5% CI", Number.Abs, Int64.Type}}),
21      #"Inserted Absolute Value" = Table.AddColumn(#"Calculated Absolute Value",
22  "Absolute Value", each Number.Abs([#"Events/10000"]), type number),
23      #"Inserted Absolute Value1" = Table.AddColumn(#"Inserted Absolute Value",
24  "Absolute Value.1", each Number.Abs([#"Events/10000 2.5% CI"]), type number),
25      #"Inserted Absolute Value2" = Table.AddColumn(#"Inserted Absolute Value1",
26  "Absolute Value.2", each Number.Abs([#"Events/10000 97.5% CI"]), type number),
27      #"Renamed Columns9" = Table.RenameColumns(#"Inserted Absolute
28  Value2",{{"Absolute Value.2", "Events/10000 97.5% CI a"}}),
29      #"Renamed Columns8" = Table.RenameColumns(#"Renamed
30  Columns9",{{"Absolute Value.1", "Events/10000 2.5% CI a"}}),
31      #"Renamed Columns7" = Table.RenameColumns(#"Renamed
32  Columns8",{{"Absolute Value", "Events/10000 a"}}),
33      #"Combine for written risk difference value" = Table.AddColumn(#"Renamed
34  Columns7", "Risk difference effect", each Text.Combine({Text.From([#"Events/10000

```

```

1  a"], "en-GB"), [RD wording], Text.From([#"Events/10000 2.5% CI a"], "en-GB"), [#"RD
2  2.5% CI wording"], Text.From([#"Events/10000 97.5% CI a"], "en-GB"), [#"RD 97.5%
3  CI wording"]}, ""), type text),
4    # "Added Conditional Column" = Table.AddColumn(# "Combine for written risk
5  difference value", "Absolute effect", each if [Risk difference effect] <> "" then [Risk
6  difference effect] else if [Effect measure] = "Hazard Ratio" then "Not applicable" else
7  if [Effect measure] = "Risk Ratio" then "Not applicable" else if [Effect measure] =
8  "Odds Ratio" then "Not applicable" else "Not applicable"),
9    # "Replaced Value1" = Table.ReplaceValue(# "Added Conditional Column", "1 more
10 events", "1 more event", Replacer.ReplaceText, {"Absolute effect"}),
11    # "Replaced Value2" = Table.ReplaceValue(# "Replaced Value1", "1 fewer
12 events", "1 fewer event", Replacer.ReplaceText, {"Absolute effect"}),
13    # "Inserted Multiplication" = Table.AddColumn(# "Replaced Value2", "Number of
14 events in control group/10000 people", each [Risk2] * 10000, type number),
15    # "Risk2 * 1000" = Table.TransformColumns(# "Inserted Multiplication", {"Risk2",
16 each _ * 1000, type number}),
17    # "Renamed Columns2" = Table.RenameColumns(# "Risk2 * 1000", {"Risk2",
18 "Number of events in control group/1000 people"}),
19    # "Changed Type2" = Table.TransformColumnTypes(# "Renamed
20 Columns2", {"Number of events in control group/1000 people", Int64.Type}, {"Number
21 of events in control group/10000 people", Int64.Type}),
22    # "Removed Columns2" = Table.RemoveColumns(# "Changed Type2", {"Sign RD",
23 "Sign RD 2.5%", "Sign RD 97.5%"}),
24    # "Total events/1000" = Table.AddColumn(# "Removed Columns2", "Total events
25 after treatment", each [# "Number of events in control group/1000 people"] +
26 [# "Events/1000 direction"], Int64.Type),
27    # "Total events/10000" = Table.AddColumn(# "Total events/1000", "Total events
28 after treatment/10000", each [# "Number of events in control group/10000 people"] +
29 [# "Events/10000"], type number),
30    # "Determining smallest number of events/1000" = Table.AddColumn(# "Total
31 events/10000", "Smallest number of events/1000", each if [Total events after
32 treatment] = null then null else if [Total events after treatment] < [# "Number of events
33 in control group/1000 people"] then [Total events after treatment] else [# "Number of
34 events in control group/1000 people"]),

```

```

1      # "Smallest number of events/10000" = Table.AddColumn(# "Determining smallest
2      number of events/1000", "Smallest number of events/10000", each if [# "Total events
3      after treatment/10000"] = null then null else if [# "Total events after treatment/10000"]
4      < [# "Number of events in control group/10000 people"] then [# "Total events after
5      treatment/10000"] else [# "Number of events in control group/10000 people"]),
6      # "Changed Type4" = Table.TransformColumnTypes(# "Smallest number of
7      events/10000", {{"Smallest number of events/1000", Int64.Type}, {"Smallest number
8      of events/10000", Int64.Type}, {"Total events after treatment/10000", Int64.Type}}),
9      # "1000" = Table.AddColumn(# "Changed Type4", "1000", each "1000"),
10     # "10000" = Table.AddColumn(# "1000", "10000", each "10000"),
11     # "Additional events from treatment/1000" = Table.AddColumn(# "10000",
12     "Additional events from treatment", each [Total events after treatment] - [# "Smallest
13     number of events/1000"], type number),
14     # "Additional events from treatment/10000" = Table.AddColumn(# "Additional
15     events from treatment/1000", "Additional events from treatment/10000", each [# "Total
16     events after treatment/10000"] - [# "Smallest number of events/10000"], Int64.Type),
17     # "Events reduced from treatment/1000" = Table.AddColumn(# "Additional events
18     from treatment/10000", "Events reduced from treatment", each [# "Number of events
19     in control group/1000 people"] - [# "Smallest number of events/1000"], type number),
20     # "Events reduced from treatment/10000" = Table.AddColumn(# "Events reduced
21     from treatment/1000", "Events reduced from treatment/10000", each [# "Number of
22     events in control group/10000 people"] - [# "Smallest number of events/10000"],
23     Int64.Type),
24     # "Changed Type3" = Table.TransformColumnTypes(# "Events reduced from
25     treatment/10000", {{"1000", Int64.Type}, {"Smallest number of events/1000",
26     Int64.Type}, {"Additional events from treatment", Int64.Type}, {"Events reduced from
27     treatment", Int64.Type}, {"10000", Int64.Type}}),
28     # "Determining largest number of events/1000" = Table.AddColumn(# "Changed
29     Type3", "Largest number of events/1000", each if [# "Number of events in control
30     group/1000 people"] = null then null else if [# "Number of events in control group/1000
31     people"] >= [Total events after treatment] then [# "Number of events in control
32     group/1000 people"] else [Total events after treatment]),
33     # "Largest number of events/10000" = Table.AddColumn(# "Determining largest
34     number of events/1000", "Largest number of events/10000", each if [# "Number of

```

```

1  events in control group/10000 people"] = null then null else if [#"Number of events in
2  control group/10000 people"] >= [#"Total events after treatment/10000"] then
3  [#"Number of events in control group/10000 people"] else [#"Total events after
4  treatment/10000"]),
5      #"Changed Type5" = Table.TransformColumnTypes(#"Largest number of
6  events/10000",{{"Largest number of events/1000", Int64.Type}, {"Largest number of
7  events/10000", Int64.Type}}),
8      #"People without event/1000" = Table.AddColumn(#"Changed Type5", "People
9  without event", each [1000] - [#"Largest number of events/1000"], Int64.Type),
10     #"People without event/10000" = Table.AddColumn(#"People without event/1000",
11     "People without event/10000", each [10000] - [#"Largest number of events/10000"],
12     Int64.Type),
13     #"Renamed Columns3" = Table.RenameColumns(#"People without
14     event/10000",{{"Smallest number of events/1000", "People with events"}, {"Events
15     reduced from treatment", "Fewer events after treatment"}, {"Smallest number of
16     events/10000", "People with events/10000"}, {"Events reduced from
17     treatment/10000", "Fewer events after treatment/10000"}}),
18     #"Added Custom1" = Table.AddColumn(#"Renamed Columns3", "|", each "|"),
19     #"Filtered Rows" = Table.SelectRows(#"Added Custom1", each ([Analysis group]
20     <> 5 and [Analysis group] <> 6)),
21     #"Added Custom2" = Table.AddColumn(#"Filtered Rows", "(", each "("),
22     #"Added Custom3" = Table.AddColumn(#"Added Custom2", ")", each ")"),
23     #"Rounded Off" = Table.TransformColumns(#"Added Custom3",{{"Mean", each
24     Number.Round(_, 5), type number}, {"CI start", each Number.Round(_, 5), type
25     number}, {"CI end", each Number.Round(_, 5), type number}}),
26     #"Inserted Merged Column1" = Table.AddColumn(#"Rounded Off", "Effect size",
27     each Text.Combine({Text.From([Mean], "en-GB"), [# "("], " "), type text),
28     #"Inserted Merged Column2" = Table.AddColumn(#"Inserted Merged Column1",
29     "Effect Size.1", each Text.Combine({[Effect size], Text.From([CI start], "en-GB"), ""},
30     type text),
31     #"Inserted Merged Column3" = Table.AddColumn(#"Inserted Merged Column2",
32     "Effect size.2", each Text.Combine({[Effect Size.1], Text.From([CI end], "en-GB"), ",
33     "), type text),

```



```

1      #"Inserted Merged Column4" = Table.AddColumn(#"Inserted Merged Column3",
2      "Effect size.3", each Text.Combine({[Effect size.2], [#]"")}, ""), type text),
3      #"Removed Columns1" = Table.RemoveColumns(#"Inserted Merged
4      Column4",{(" ", "Effect size", "Effect Size.1", "Effect size.2"}),
5      #"Renamed Columns5" = Table.RenameColumns(#"Removed Columns1",{{"Effect
6      size.3", "Effect size"}}),
7      #"Duplicated Column" = Table.DuplicateColumn(#"Renamed Columns5", "Analysis
8      name", "Analysis name - Copy"),
9      #"Split Column by Delimiter" = Table.SplitColumn(#"Duplicated Column", "Analysis
10     name - Copy", Splitter.SplitTextByDelimiter("(", QuoteStyle.Csv), {"Analysis name -
11     Copy.1", "Analysis name - Copy.2"}),
12     #"Changed Type10" = Table.TransformColumnTypes(#"Split Column by
13     Delimiter",{{"Effect T", type text}, {"Analysis name - Copy.1", type text}, {"Analysis
14     name - Copy.2", type text}}),
15     #"Removed Columns4" = Table.RemoveColumns(#"Changed Type10",{Analysis
16     name - Copy.1"}),
17     #"Extracted Text Before Delimiter" = Table.TransformColumns(#"Removed
18     Columns4", {{"Analysis name - Copy.2", each Text.BeforeDelimiter(_, "("), type
19     text}}),
20     #"Renamed Columns6" = Table.RenameColumns(#"Extracted Text Before
21     Delimiter",{{"Analysis name - Copy.2", "Scale"}}),
22     #"Filtered Rows1" = Table.SelectRows(#"Renamed Columns6", each ([Scale] <>
23     "risk difference")),
24     #"Replaced Value3" = Table.ReplaceValue(#"Filtered Rows1","Pregnancy -
25     ", "", Replacer.ReplaceText,{"Analysis group name"}),
26     #"Replaced Value4" = Table.ReplaceValue(#"Replaced Value3","Not pregnant -
27     ", "", Replacer.ReplaceText,{"Analysis group name"}),
28     #"Added Conditional Column1" = Table.AddColumn(#"Replaced Value4", "MID
29     multiplier", each if [Outcome] = "All-cause mortality" then 10 else if [Outcome] =
30     "Thromboembolic events after surgery" then 0.2 else 1),
31     #"Inserted Multiplication4" = Table.AddColumn(#"Added Conditional Column1",
32     "Multiplication", each [Median control group rate] * [MID multiplier] * 1000, type
33     number),

```

```

1      #"Rounded Up" = Table.TransformColumns(#"Inserted
2 Multiplication4",{{"Multiplication", Number.RoundUp, Int64.Type}}),
3      #"Added Conditional Column2" = Table.AddColumn(#"Rounded Up", "MID", each if
4 [Outcome] = "All-cause mortality" then 10 else [Multiplication]),
5      #"Inserted Multiplication5" = Table.AddColumn(#"Added Conditional Column2",
6 "Negative MID", each [MID] * -1, type number),
7      #"Added Conditional Column3" = Table.AddColumn(#"Inserted Multiplication5",
8 "Positive MID side", each if [#"Events/10000"] = null then null else if
9 [#"Events/10000"] > [MID] then 1 else if [#"Events/10000"] = [MID] then 0 else if
10 [#"Events/10000"] < [MID] then -1 else null),
11      #"Added Conditional Column4" = Table.AddColumn(#"Added Conditional
12 Column3", "Negative MID side", each if [#"Events/1000 direction"] = null then null
13 else if [#"Events/10000"] < [Negative MID] then -1 else if [#"Events/10000"] =
14 [Negative MID] then 0 else if [#"Events/10000"] > [Negative MID] then 1 else null),
15      #"Inserted Merged Column5" = Table.AddColumn(#"Added Conditional Column4",
16 "MID code", each Text.Combine({Text.From([Positive MID side], "en-GB"),
17 Text.From([Negative MID side], "en-GB")}, "-"), type text),
18      #"Replaced Value5" = Table.ReplaceValue(#"Inserted Merged
19 Column5","",null,Replacer.ReplaceValue,{"MID code"}),
20      #"Added Conditional Column5" = Table.AddColumn(#"Replaced Value5", "Clinical
21 importance", each if [MID code] = "-1--1" then "Clinically important benefit" else if
22 [MID code] = "-1-0" then "Clinically important benefit" else if [MID code] = "1-1" then
23 "Clinically important harm" else if [MID code] = "-1-1" then "No clinically important
24 difference" else if [MID code] = "0-1" then "Clinically important harm" else null),
25      #"Inserted Merged Column6" = Table.AddColumn(#"Added Conditional Column5",
26 "Absolute effect (clinical importance)", each Text.Combine({[Absolute effect], [Clinical
27 importance]}), "
28 "), type text),
29      #"MID absolute" = Table.AddColumn(#"Inserted Merged Column6", "MID
30 absolute", each [MID], type number),
31      #"Added Prefix" = Table.TransformColumns(#"MID absolute", {{"MID absolute",
32 each "MID (clinical importance) = " & Text.From(_, "en-GB"), type text}}),
33      #"Added Suffix" = Table.TransformColumns(#"Added Prefix", {{"MID absolute",
34 each _ & " events per 10,000 people", type text}})

```

1 in

2 #"Added Suffix"

3 **Risk of bias**

4 let

5 Source = Csv.Document(Web.Contents("source"),[Delimiter=","]),

6 #"Promoted Headers" = Table.PromoteHeaders(Source,

7 [PromoteAllScalars=true]),

8 #"Changed Type" = Table.TransformColumnTypes(#"Promoted

9 Headers",{{"Analysis group", Int64.Type}, {"Analysis number", Int64.Type}, {"Analysis

10 name", type text}, {"Subgroup", type any}, {"Applicability", type text}, {"Study", type

11 text}, {"Study year", Int64.Type}, {"GIV Mean", type number}, {"GIV SE", type

12 number}, {"Experimental mean", type number}, {"Experimental SD", type number},

13 {"Experimental cases", Int64.Type}, {"Experimental N", Int64.Type}, {"Control mean",

14 type number}, {"Control SD", type number}, {"Control cases", Int64.Type}, {"Control

15 N", Int64.Type}, {"O-E", type any}, {"Variance", type any}, {"Weight", type number},

16 {"Mean", type number}, {"CI start", type number}, {"CI end", type number},

17 {"Footnotes", type any}}),

18 #"Splitting by "" at "" to separate follow up time" = Table.SplitColumn(#"Changed

19 Type", "Analysis name", Splitter.SplitTextByDelimiter(" at ", QuoteStyle.Csv),

20 {"Analysis name.1", "Analysis name.2"}),

21 #"Changed Type1" = Table.TransformColumnTypes(#"Splitting by "" at "" to

22 separate follow up time",{{"Analysis name.1", type text}, {"Analysis name.2", type

23 text}}),

24 #"Renamed Columns" = Table.RenameColumns(#"Changed Type1",{{"Analysis

25 name.2", "Follow up time"}}),

26 #"Splitting by "" (" to separate scale information" = Table.SplitColumn(#"Renamed

27 Columns", "Analysis name.1", Splitter.SplitTextByDelimiter(" (", QuoteStyle.Csv),

28 {"Analysis name.1.1", "Analysis name.1.2"}),

29 #"Changed Type2" = Table.TransformColumnTypes(#"Splitting by "" (" to

30 separate scale information",{{"Analysis name.1.1", type text}, {"Analysis name.1.2",

31 type text}}),

32 #"Renamed Columns1" = Table.RenameColumns(#"Changed Type2",{{"Analysis

33 name.1.2", "Scale"}, {"Analysis name.1.1", "Protocol Outcome"}}),

```

1      #"Removing closing bracket" = Table.ReplaceValue(#"Renamed
2 Columns1", ",", "", Replacer.ReplaceText, {"Scale"}),
3      #"Making short outcome" = Table.AddColumn(#"Removing closing bracket",
4 "Outcome short", each if [Protocol Outcome] = "Thromboembolic events after
5 surgery" then "Thromboembolic events" else if [Protocol Outcome] = "All-cause
6 mortality" then "Mortality" else if [Protocol Outcome] = "Pulmonary embolism" then
7 "PE" else if [Protocol Outcome] = "Deep vein thrombosis" then "DVT" else if [Protocol
8 Outcome] = "Myocardial infarction" then "MI" else if [Protocol Outcome] = "Ischaemic
9 stroke" then "Stroke" else if [Protocol Outcome] = "Infection" then "Infection" else if
10 [Protocol Outcome] = "All-cause readmission" then "Readmission" else if [Protocol
11 Outcome] = "Seizures" then "Seizures" else if [Protocol Outcome] = "Reoperation"
12 then "Reoperation" else "?"),
13      #"Replaced Value2" = Table.ReplaceValue(#"Making short outcome", "risk
14 ratio", null, Replacer.ReplaceValue, {"Scale"}),
15      #"Filtered Rows1" = Table.SelectRows(#"Replaced Value2", each ([Scale] <> "risk
16 difference")),
17      #"Inserted Merged Column1" = Table.AddColumn(#"Filtered Rows1", "Study ID-
18 Outcome-Scale", each Text.Combine({[Study ID], [Protocol Outcome], [Scale]}, "-"),
19 type text),
20      #"Merged Queries" = Table.NestedJoin(#"Inserted Merged Column1", {"Study ID"},
21 #"Study level data", {"Study ID"}, "BiasInd", JoinKind.LeftOuter),
22      #"Expanded Critical appraisal no edits" = Table.ExpandTableColumn(#"Merged
23 Queries", "BiasInd", {"Follow-up", "Short population", "Short comparison"}, {"Follow-
24 up", "Short population", "Short comparison"}),
25      #"Renamed Columns2" = Table.RenameColumns(#"Expanded Critical appraisal
26 no edits", {"Follow-up", "Timepoint (days)"}),
27      #"Merged Queries1" = Table.NestedJoin(#"Renamed Columns2", {"Study ID-
28 Outcome-Scale"}, BiasInd, {"Study ID-Outcome-Scale"}, "BiasInd",
29 JoinKind.LeftOuter),
30      #"Expanded BiasInd" = Table.ExpandTableColumn(#"Merged Queries1",
31 "BiasInd", {"Overall risk of bias", "Overall directness"}, {"Overall risk of bias", "Overall
32 directness"}),
33      #"Removed Duplicates" = Table.Distinct(#"Expanded BiasInd"),

```

```

1      #"Removed Other Columns" = Table.SelectColumns(#"Removed
2 Duplicates",{ "Analysis group", "Analysis number", "Protocol Outcome", "Scale",
3 "Subgroup", "Subgroup number", "Applicability", "Study", "Study ID", "Weight",
4 "Outcome short", "Timepoint (days)", "Short population", "Short comparison", "Overall
5 risk of bias", "Overall directness"}),
6      #"Reordered Columns" = Table.ReorderColumns(#"Removed Other
7 Columns",{ "Analysis group", "Analysis number", "Study", "Protocol Outcome",
8 "Scale", "Outcome short", "Short population", "Short comparison", "Timepoint (days)",
9 "Subgroup", "Applicability", "Weight", "Overall risk of bias", "Overall directness"}),
10     #"Serious to Moderate" = Table.ReplaceValue(#"Reordered
11 Columns","Serious","Moderate",Replacer.ReplaceText,{"Overall risk of bias"}),
12     #"Very Serious to High" = Table.ReplaceValue(#"Serious to Moderate","Very
13 Serious","High",Replacer.ReplaceText,{"Overall risk of bias"}),
14     #"Changing RoB from text to numbers" = Table.AddColumn(#"Very Serious to
15 High", "Risk of bias number", each if [Overall risk of bias] = "Low" then 0 else if
16 [Overall risk of bias] = "Moderate" then 1 else if [Overall risk of bias] = "High" then 2
17 else "?"),
18     #"Removed Other Columns1" = Table.SelectColumns(#"Changing RoB from text
19 to numbers",{ "Analysis group", "Analysis number", "Protocol Outcome", "Scale",
20 "Subgroup number", "Outcome short", "Study ID", "Weight", "Overall risk of bias",
21 "Overall directness", "Risk of bias number"}),
22     #"Replaced Value" = Table.ReplaceValue(#"Removed Other
23 Columns1",0,100,Replacer.ReplaceValue,{"Weight"}),
24     #"Changing weight null to weight 100" = Table.ReplaceValue(#"Replaced
25 Value",null,100,Replacer.ReplaceValue,{"Weight"}),
26     #"Calculating weighted RoB" = Table.AddColumn(#"Changing weight null to weight
27 100", "Weighted RoB", each [Risk of bias number] * [Weight], type number),
28     #"Dividing by 100" = Table.TransformColumns(#"Calculating weighted RoB",
29 {"Weighted RoB", each _ / 100, type number}),
30     #"Removed Other Columns2" = Table.SelectColumns(#"Dividing by
31 100",{ "Analysis group", "Analysis number", "Protocol Outcome", "Scale", "Weighted
32 RoB"}),
33     #"Duplicated Column" = Table.DuplicateColumn(#"Removed Other Columns2",
34 "Protocol Outcome", "Protocol Outcome - Copy"),

```

```

1      #"Pivoted Column to calculate combined RoB" = Table.Pivot(#"Duplicated
2      Column", List.Distinct(#"Duplicated Column"[#"Protocol Outcome - Copy"]), "Protocol
3      Outcome - Copy", "Weighted RoB", List.Sum),
4      #"Merged Columns1" =
5      Table.CombineColumns(Table.TransformColumnTypes(#"Pivoted Column to
6      calculate combined RoB", {"All-cause mortality", type text}, {"Thromboembolic
7      events after surgery", type text}, {"Pulmonary embolism", type text}, {"Deep vein
8      thrombosis", type text}, {"Myocardial infarction", type text}, {"Ischaemic stroke", type
9      text}, {"Infection", type text}, {"All-cause readmission", type text}, {"Seizures", type
10     text}, {"Reoperation", type text}}, "en-GB"), {"All-cause mortality", "Thromboembolic
11     events after surgery", "Pulmonary embolism", "Deep vein thrombosis", "Myocardial
12     infarction", "Ischaemic stroke", "Infection", "All-cause readmission", "Seizures",
13     "Reoperation"}, Combiner.CombineTextByDelimiter("", QuoteStyle.None), "Summed
14     weighted RoB Number"),
15     #"Changed Type3" = Table.TransformColumnTypes(#"Merged
16     Columns1", {"Summed weighted RoB Number", type number}),
17     #"Calculation of RoB" = Table.AddColumn(#"Changed Type3", "Outcome RoB",
18     each if [Summed weighted RoB Number] < 0 then "?" else if [Summed weighted RoB
19     Number] <= 0.66 then "Not serious" else if [Summed weighted RoB Number] <= 1.32
20     then "Serious" else if [Summed weighted RoB Number] <= 2 then "Very serious" else
21     "?"),
22     #"Analyst manual edit of RoB" = Table.AddColumn(#"Calculation of RoB", "Edited
23     outcome RoB", each if [Protocol Outcome] = "Falls" then "Serious" else null),
24     #"Risk of Bias final determination" = Table.AddColumn(#"Analyst manual edit of
25     RoB", "RoB", each if [Edited outcome RoB] <> null then [Edited outcome RoB] else
26     [Outcome RoB]),
27     #"Merged Queries2" = Table.NestedJoin(#"Risk of Bias final determination",
28     {"Analysis group", "Analysis number"}, #"Bias method", {"Analysis group", "Analysis
29     number"}, "Bias method", JoinKind.LeftOuter),
30     #"Expanded Bias method" = Table.ExpandTableColumn(#"Merged Queries2",
31     "Bias method", {"Method"}, {"Method"}),
32     #"Risk of bias reason determination" = Table.AddColumn(#"Expanded Bias
33     method", "RoB reason", each if [RoB] = "Serious" then "Risk of bias: Downgraded
34     once. Serious risk of bias in the evidence contributing to the outcomes. More than

```

```

1  50% of the weight of the evidence came from studies at moderate or high risk of bias
2  as per" else if [RoB] = "Very serious" then "Risk of bias: Downgraded twice. Very
3  serious risk of bias in the evidence contributing to the outcomes. More than 50% of
4  the weight of the evidence came from studies at high risk of bias as per" else null),
5    #"Added Conditional Column" = Table.AddColumn(#"Risk of bias reason
6  determination", "Method merge", each if [RoB reason] = null then null else [Method]),
7    #"Merged Columns" = Table.CombineColumns(#"Added Conditional
8  Column",{ "RoB reason", "Method merge"},Combiner.CombineTextByDelimiter(" ",
9  QuoteStyle.None),"RoB reason"),
10   #"Replaced Value1" = Table.ReplaceValue(#"Merged Columns",
11   ",null,Replacer.ReplaceValue,{"RoB reason"}),
12   #"Inserted Merged Column" = Table.AddColumn(#"Replaced Value1", "GRADE
13   code", each Text.Combine({Text.From([Analysis group], "en-GB"),
14   Text.From([Analysis number], "en-GB")}, "-"), type text),
15   #"Removed Other Columns3" = Table.SelectColumns(#"Inserted Merged
16   Column",{ "Analysis group", "Analysis number", "RoB", "RoB reason", "GRADE
17   code"}),
18   #"Reordered Columns2" = Table.ReorderColumns(#"Removed Other
19   Columns3",{ "Analysis group", "Analysis number", "GRADE code", "RoB", "RoB
20   reason"}),
21   #"Filtered Rows" = Table.SelectRows(#"Reordered Columns2", each ([Analysis
22   group] <> 5 and [Analysis group] <> 6))
23   in
24   #"Filtered Rows"

```

25 Bias method

```

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

```

```

1  number}, {"Experimental mean", type number}, {"Experimental SD", type number},
2  {"Experimental cases", Int64.Type}, {"Experimental N", Int64.Type}, {"Control mean",
3  type number}, {"Control SD", type number}, {"Control cases", Int64.Type}, {"Control
4  N", Int64.Type}, {"O-E", type any}, {"Variance", type any}, {"Weight", type number},
5  {"Mean", type number}, {"CI start", type number}, {"CI end", type number},
6  {"Footnotes", type any}}),
7    #"Splitting by "" at "" to separate follow up time" = Table.SplitColumn(#"Changed
8  Type", "Analysis name", Splitter.SplitTextByDelimiter(" at ", QuoteStyle.Csv),
9  {"Analysis name.1", "Analysis name.2"}),
10   #"Changed Type1" = Table.TransformColumnTypes(#"Splitting by "" at "" to
11  separate follow up time",{{"Analysis name.1", type text}, {"Analysis name.2", type
12  text}}),
13   #"Renamed Columns" = Table.RenameColumns(#"Changed Type1",{{"Analysis
14  name.2", "Follow up time"}}),
15   #"Splitting by "" (" to separate scale information" = Table.SplitColumn(#"Renamed
16  Columns", "Analysis name.1", Splitter.SplitTextByDelimiter(" (", QuoteStyle.Csv),
17  {"Analysis name.1.1", "Analysis name.1.2"}),
18   #"Changed Type2" = Table.TransformColumnTypes(#"Splitting by "" (" to
19  separate scale information",{{"Analysis name.1.1", type text}, {"Analysis name.1.2",
20  type text}}),
21   #"Renamed Columns1" = Table.RenameColumns(#"Changed Type2",{{"Analysis
22  name.1.2", "Scale"}, {"Analysis name.1.1", "Protocol Outcome"}}),
23   #"Removing closing bracket" = Table.ReplaceValue(#"Renamed
24  Columns1",")","",Replacer.ReplaceText,{"Scale"}),
25   #"Making short outcome" = Table.AddColumn(#"Removing closing bracket",
26  "Outcome short", each if [Protocol Outcome] = "Thromboembolic events after
27  surgery" then "Thromboembolic events" else if [Protocol Outcome] = "All-cause
28  mortality" then "Mortality" else if [Protocol Outcome] = "Pulmonary embolism" then
29  "PE" else if [Protocol Outcome] = "Deep vein thrombosis" then "DVT" else if [Protocol
30  Outcome] = "Myocardial infarction" then "MI" else if [Protocol Outcome] = "Ischaemic
31  stroke" then "Stroke" else if [Protocol Outcome] = "Infection" then "Infection" else if
32  [Protocol Outcome] = "All-cause readmission" then "Readmission" else if [Protocol
33  Outcome] = "Seizures" then "Seizures" else if [Protocol Outcome] = "Reoperation"
34  then "Reoperation" else "?"),

```



```

1      #"Merged Queries" = Table.NestedJoin(#"Making short outcome", {"Study ID"},
2      #"Study level data", {"Study ID"}, "BiasInd", JoinKind.LeftOuter),
3      #"Expanded Critical appraisal no edits" = Table.ExpandTableColumn(#"Merged
4      Queries", "BiasInd", {"Follow-up", "Short population", "Short comparison"}, {"Follow-
5      up", "Short population", "Short comparison"}),
6      #"Renamed Columns2" = Table.RenameColumns(#"Expanded Critical appraisal
7      no edits",{"Follow-up", "Timepoint (days)"}),
8      #"Merged Queries1" = Table.NestedJoin(#"Renamed Columns2", {"Study ID"},
9      "Protocol Outcome", BiasInd, {"Study ID", "Protocol Outcome"}, "BiasInd",
10     JoinKind.LeftOuter),
11     #"Expanded BiasInd" = Table.ExpandTableColumn(#"Merged Queries1",
12     "BiasInd", {"Method"}, {"Method"}),
13     #"Removed Duplicates" = Table.Distinct(#"Expanded BiasInd"),
14     #"Removed Other Columns" = Table.SelectColumns(#"Removed
15     Duplicates",{"Analysis group", "Analysis number", "Protocol Outcome", "Scale",
16     "Method"}),
17     #"Reordered Columns" = Table.ReorderColumns(#"Removed Other
18     Columns",{"Analysis group", "Analysis number", "Protocol Outcome", "Scale",
19     "Method"}),
20     #"Duplicated Column" = Table.DuplicateColumn(#"Reordered Columns", "Protocol
21     Outcome", "Protocol Outcome - Copy"),
22     #"Method number conversion" = Table.AddColumn(#"Duplicated Column",
23     "Method number", each if [Method] = "ROB 2.0" then 1 else if [Method] = "ROBINS-I"
24     then 2 else if [Method] = "ROBIS" then 3 else "?"),
25     #"Changed Type4" = Table.TransformColumnTypes(#"Method number
26     conversion",{"Method number", type number}),
27     #"Removed Columns" = Table.RemoveColumns(#"Changed Type4",{"Method"}),
28     #"Pivoted Column to calculate combined methods" = Table.Pivot(#"Removed
29     Columns", List.Distinct(#"Removed Columns"[#"Protocol Outcome - Copy"]),
30     "Protocol Outcome - Copy", "Method number", List.Average),
31     #"Merged Columns1" =
32     Table.CombineColumns(Table.TransformColumnTypes(#"Pivoted Column to
33     calculate combined methods", {"All-cause mortality", type text}, {"Thromboembolic
34     events after surgery", type text}, {"Pulmonary embolism", type text}, {"Deep vein

```

```

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

```

16 Indirectness

```

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

```

```

1      #"Changed Type1" = Table.TransformColumnTypes(#"Splitting by "" at "" to
2      separate follow up time",{{"Analysis name.1", type text}, {"Analysis name.2", type
3      text}}),
4      #"Renamed Columns" = Table.RenameColumns(#"Changed Type1",{{"Analysis
5      name.2", "Follow up time"}}),
6      #"Splitting by "" (" to separate scale information" = Table.SplitColumn(#"Renamed
7      Columns", "Analysis name.1", Splitter.SplitTextByDelimiter("(", QuoteStyle.Csv),
8      {"Analysis name.1.1", "Analysis name.1.2"}),
9      #"Changed Type2" = Table.TransformColumnTypes(#"Splitting by "" (" to
10     separate scale information",{{"Analysis name.1.1", type text}, {"Analysis name.1.2",
11     type text}}),
12     #"Renamed Columns1" = Table.RenameColumns(#"Changed Type2",{{"Analysis
13     name.1.2", "Scale"}, {"Analysis name.1.1", "Protocol Outcome"}}),
14     #"Removing closing bracket" = Table.ReplaceValue(#"Renamed
15     Columns1",")","",Replacer.ReplaceText,{"Scale"}),
16     #"Making short outcome" = Table.AddColumn(#"Removing closing bracket",
17     "Outcome short", each if [Protocol Outcome] = "Thromboembolic events after
18     surgery" then "Thromboembolic events" else if [Protocol Outcome] = "All-cause
19     mortality" then "Mortality" else if [Protocol Outcome] = "Pulmonary embolism" then
20     "PE" else if [Protocol Outcome] = "Deep vein thrombosis" then "DVT" else if [Protocol
21     Outcome] = "Myocardial infarction" then "MI" else if [Protocol Outcome] = "Ischaemic
22     stroke" then "Stroke" else if [Protocol Outcome] = "Infection" then "Infection" else if
23     [Protocol Outcome] = "All-cause readmission" then "Readmission" else if [Protocol
24     Outcome] = "Readmission due to thrombosis" then "Thrombosis readmission" else
25     "?"),
26     #"Filtered Rows1" = Table.SelectRows(#"Making short outcome", each ([Scale] <>
27     "risk difference")),
28     #"Replaced Value2" = Table.ReplaceValue(#"Filtered Rows1","risk
29     ratio",null,Replacer.ReplaceValue,{"Scale"}),
30     #"Inserted Merged Column2" = Table.AddColumn(#"Replaced Value2", "Study ID-
31     Outcome-Scale", each Text.Combine({[Study ID], [Protocol Outcome], [Scale]}, "-"),
32     type text),

```

```

1      #"Merged Queries" = Table.NestedJoin(#"Inserted Merged Column2", {"Study ID-
2 Outcome-Scale"}, BiasInd, {"Study ID-Outcome-Scale"}, "BiasInd",
3 JoinKind.LeftOuter),
4      #"Expanded BiasInd" = Table.ExpandTableColumn(#"Merged Queries", "BiasInd",
5 {"Overall risk of bias", "Overall directness"}, {"Overall risk of bias", "Overall
6 directness"}),
7      #"Removed Duplicates" = Table.Distinct(#"Expanded BiasInd"),
8      #"Reordered Columns1" = Table.ReorderColumns(#"Removed
9 Duplicates",{"Analysis group", "Analysis number", "Study", "Protocol Outcome",
10 "Scale", "Outcome short", "Subgroup", "Applicability", "Study year", "Weight", "Overall
11 risk of bias", "Overall directness"}),
12      #"Removed Other Columns" = Table.SelectColumns(#"Reordered
13 Columns1",{"Analysis group", "Analysis number", "Study", "Protocol Outcome",
14 "Follow up time", "Scale", "Subgroup number", "Outcome short", "Subgroup", "Study
15 ID", "Applicability", "Weight", "Overall risk of bias", "Overall directness"}),
16      #"Fully applicable to Directly applicable" = Table.ReplaceValue(#"Removed Other
17 Columns", "Fully applicable", "Directly applicable", Replacer.ReplaceText, {"Overall
18 directness"}),
19      #"Changing directness from text to numbers" = Table.AddColumn(#"Fully
20 applicable to Directly applicable", "Directness number", each if [Overall directness] =
21 "Directly applicable" then 0 else if [Overall directness] = "Partially applicable" then 1
22 else if [Overall directness] = "Indirectly applicable" then 2 else "?"),
23      #"Removed Columns3" = Table.RemoveColumns(#"Changing directness from text
24 to numbers", {"Study", "Subgroup", "Applicability"}),
25      #"Replaced Value" = Table.ReplaceValue(#"Removed
26 Columns3", 0, 100, Replacer.ReplaceValue, {"Weight"}),
27      #"Sorted Rows" = Table.Sort(#"Replaced Value", {"Analysis group",
28 Order.Ascending}, {"Analysis number", Order.Ascending}),
29      #"Calculating weighted directness" = Table.AddColumn(#"Sorted Rows",
30 "Weighted directness", each [Directness number] * [Weight], type number),
31      #"Dividing by 100" = Table.TransformColumns(#"Calculating weighted directness",
32 {"Weighted directness", each _ / 100, type number}),

```

```

1      #"Removed Columns4" = Table.RemoveColumns(#"Dividing by 100",{ "Weight",
2      "Overall risk of bias", "Overall directness", "Directness number", "Outcome short",
3      "Study ID", "Subgroup number"}),
4      #"Duplicated Column" = Table.DuplicateColumn(#"Removed Columns4", "Protocol
5      Outcome", "Protocol Outcome - Copy"),
6      #"Pivoted Column to calculate combined directness" = Table.Pivot(#"Duplicated
7      Column", List.Distinct(#"Duplicated Column"[#"Protocol Outcome - Copy"]), "Protocol
8      Outcome - Copy", "Weighted directness", List.Sum),
9      #"Merged Columns" =
10     Table.CombineColumns(Table.TransformColumnTypes(#"Pivoted Column to
11     calculate combined directness", {"All-cause mortality", type text}, {"Thromboembolic
12     events after surgery", type text}, {"Pulmonary embolism", type text}, {"Deep vein
13     thrombosis", type text}, {"Myocardial infarction", type text}, {"Ischaemic stroke", type
14     text}, {"Infection", type text}, {"All-cause readmission", type text}, {"Seizures", type
15     text}, {"Reoperation", type text}}, "en-GB"), {"All-cause mortality", "Thromboembolic
16     events after surgery", "Pulmonary embolism", "Deep vein thrombosis", "Myocardial
17     infarction", "Ischaemic stroke", "Infection", "All-cause readmission", "Seizures",
18     "Reoperation"}, Combiner.CombineTextByDelimiter("", QuoteStyle.None), "Summed
19     weighted directness number"),
20     #"Changed Type3" = Table.TransformColumnTypes(#"Merged
21     Columns", {"Summed weighted directness number", type number}),
22     #"Calculation of directness" = Table.AddColumn(#"Changed Type3", "Outcome
23     directness", each if [Summed weighted directness number] < 0 then "?" else if
24     [Summed weighted directness number] <= 0.66 then "Not serious" else if [Summed
25     weighted directness number] <= 1.32 then "Serious" else if [Summed weighted
26     directness number] <= 2 then "Very serious" else "?"),
27     #"Analyst manual edit of directness" = Table.AddColumn(#"Calculation of
28     directness", "Edited outcome directness", each if [Protocol Outcome] = "Falls" then
29     null else null),
30     #"Directness final determination" = Table.AddColumn(#"Analyst manual edit of
31     directness", "Indirectness", each if [Edited outcome directness] <> null then [Edited
32     outcome directness] else [Outcome directness]),

```

```

1      #"Inserted Merged Column1" = Table.AddColumn(#"Directness final
2      determination", "Indirectness-outcome", each Text.Combine({[Indirectness], [Protocol
3      Outcome]}, "-"), type text),
4      #"Replaced Value1" = Table.ReplaceValue(#"Inserted Merged Column1", "Not
5      serious", "IGNORE", Replacer.ReplaceText, {"Indirectness-outcome"}),
6      #"Indirectness reason determination" = Table.AddColumn(#"Replaced Value1",
7      "Indirectness reason", each if Text.Contains([#"Indirectness-outcome"], "IGNORE")
8      then null else if [#"Indirectness-outcome"] = "Very serious-Ischaemic stroke" then
9      "Indirectness: Downgraded twice. Very serious indirectness due to >50% of overall
10     weighting indirect. Outcome indirectness as people could have had haemorrhagic
11     strokes as well as ischaemic strokes and at least one study has less than 500
12     participants in at least one study arm." else if [#"Indirectness-outcome"] = "Serious-
13     Ischaemic stroke" then "Indirectness: Downgraded once. Serious indirectness due to
14     >50% of overall weighting partially direct or indirect. Outcome indirectness as people
15     could have had haemorrhagic strokes as well as ischaemic strokes." else if
16     [#"Indirectness-outcome"] = "Very serious-Thromboembolic events after surgery"
17     then "Indirectness: Downgraded twice. Very serious indirectness due to >50% of
18     overall weighting indirect. Outcome indirectness as thromboembolic events after
19     surgery aggregate is not consistently including same events and at least one study
20     has less than 500 participants in at least one study arm." else if [#"Indirectness-
21     outcome"] = "Serious-Thromboembolic events after surgery" then "Indirectness:
22     Downgraded once. Serious indirectness due to >50% of overall weighting partially
23     direct or indirect. Outcome indirectness as thromboembolic events after surgery
24     aggregate is not consistently including same events." else if
25     Text.Contains([#"Indirectness-outcome"], "Serious") then "Indirectness: Downgraded
26     once. Serious indirectness due to >50% of overall weighting partially direct or
27     indirect. Outcome indirectness as at least a study contains less than 500 participants
28     in at least one arm of a study" else null),
29     #"Inserted Merged Column" = Table.AddColumn(#"Indirectness reason
30     determination", "GRADE code", each Text.Combine({Text.From([Analysis group],
31     "en-GB"), Text.From([Analysis number], "en-GB")}, "-"), type text),
32     #"Removed Columns5" = Table.RemoveColumns(#"Inserted Merged
33     Column", {"Protocol Outcome", "Scale", "Summed weighted directness number",
34     "Outcome directness", "Edited outcome directness"}),

```

```

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

```

8 Imprecision

```

9 let
10     Source = Csv.Document(Web.Contents("source"),[Delimiter=","]),
11     #"Promoted Headers" = Table.PromoteHeaders(Source,
12 [PromoteAllScalars=true]),
13     #"Changed Type" = Table.TransformColumnTypes(#"Promoted
14 Headers",{{"Analysis group", Int64.Type}, {"Analysis number", Int64.Type}, {"Analysis
15 name", type text}, {"Analysis group name", type text}, {"Data source", type text},
16 {"Data source eligibility", type text}, {"Data type", type text}, {"Log-scale data", type
17 logical}, {"Outcome", type text}, {"Intervention grouping", type text}, {"Experimental
18 intervention", type any}, {"Control intervention", type any}, {"Subgroup by", type any},
19 {"Filter criteria", type any}, {"Experimental group label", type text}, {"Control group
20 label", type text}, {"Statistical method", type text}, {"Effect measure", type text}, {"Unit
21 of effect measure", type any}, {"Analysis model", type text}, {"Heterogeneity
22 estimator", type any}, {"Tau² CI", type logical}, {"Subgroup estimates", type logical},
23 {"Overall estimates", type logical}, {"Test for subgroup differences", type logical},
24 {"Prediction interval", type logical}, {"Swap event and non-event", type logical}, {"CI
25 method", type text}, {"CI/PI level", type number}, {"Experimental cases", Int64.Type},
26 {"Experimental N", Int64.Type}, {"Control cases", Int64.Type}, {"Control N",
27 Int64.Type}, {"Mean", type number}, {"CI start", type number}, {"CI end", type
28 number}, {"PI start", type any}, {"PI end", type any}, {"Heterogeneity Tau²", type any},
29 {"Tau² CI start", type any}, {"Tau² CI end", type any}, {"Heterogeneity Chi²", type
30 number}, {"Heterogeneity df", Int64.Type}, {"Heterogeneity P", type number},
31 {"Heterogeneity I²", type number}, {"Effect Z", type number}, {"Effect P", type
32 number}, {"Subgroup Chi²", type any}, {"Subgroup df", type any}, {"Subgroup P", type
33 any}, {"Subgroup I²", type any}, {"ID", type text})),

```

```

1      #"Filtered Rows" = Table.SelectRows(#"Changed Type", each ([Analysis group] <>
2 5 and [Analysis group] <> 6)),
3      #"Removed Other Columns" = Table.SelectColumns(#"Filtered Rows",{"Analysis
4 group", "Analysis number", "Analysis name", "Analysis group name", "Outcome",
5 "Experimental intervention", "Control intervention", "Effect measure", "Experimental
6 cases", "Experimental N", "Control cases", "Control N", "Mean", "CI start", "CI end"}),
7      #"Calculating sample size" = Table.AddColumn(#"Removed Other Columns",
8 "Sample size", each [Experimental N] + [Control N], Int64.Type),
9      #"Renamed Columns" = Table.RenameColumns(#"Calculating sample
10 size",{{"Experimental N", "n1"}, {"Control N", "n2"}, {"Experimental cases", "Number of
11 events 1"}, {"Control cases", "Number of events 2"}}),
12      #"Calculating non-events 1" = Table.AddColumn(#"Renamed Columns", "Non-
13 events 1", each [n1] - [Number of events 1], Int64.Type),
14      #"Calculating non-events 2" = Table.AddColumn(#"Calculating non-events 1",
15 "Non-events 2", each [n2] - [Number of events 2], Int64.Type),
16      #"Calculating Risk1" = Table.AddColumn(#"Calculating non-events 2", "Risk1",
17 each [Number of events 1] / [n1], type number),
18      #"Calculating Risk2" = Table.AddColumn(#"Calculating Risk1", "Risk2", each
19 [Number of events 2] / [n2], type number),
20      #"Calculating risk difference" = Table.AddColumn(#"Calculating Risk2", "Risk
21 difference", each [Risk1] - [Risk2], type number),
22      #"Inserted Multiplication1" = Table.AddColumn(#"Calculating risk difference",
23 "Event1xNonevent1", each [Number of events 1] * [#"Non-events 1"], Int64.Type),
24      #"Inserted Multiplication2" = Table.AddColumn(#"Inserted Multiplication1",
25 "EventxNonevent2", each [Number of events 2] * [#"Non-events 2"], Int64.Type),
26      #"Inserted Cube" = Table.AddColumn(#"Inserted Multiplication2", "n1^3", each
27 Number.Power([#"Non-events 1"], 3), Int64.Type),
28      #"Inserted Cube1" = Table.AddColumn(#"Inserted Cube", "n2^3", each
29 Number.Power([n2], 3), Int64.Type),
30      #"Inserted Division3" = Table.AddColumn(#"Inserted Cube1",
31 "eventxnonevent1/n1^3", each [Event1xNonevent1] / [#"n1^3"], type number),
32      #"Inserted Division4" = Table.AddColumn(#"Inserted Division3", "Division", each
33 [EventxNonevent2] / [#"n2^3"], type number),

```



```

1      #"Renamed Columns1" = Table.RenameColumns(#"Inserted
2      Division4",{{"Division", "eventxnonevent2/n2^3"}}),
3      #"Inserted Addition2" = Table.AddColumn(#"Renamed Columns1",
4      "eventxnonevent1/n1^3 + eventxnonevent2/n2^3", each [#"eventxnonevent1/n1^3"] +
5      [#"eventxnonevent2/n2^3"], type number),
6      #"Calculate SE(RD)" = Table.AddColumn(#"Inserted Addition2", "SE(RD)", each
7      Number.Sqrt([#"eventxnonevent1/n1^3 + eventxnonevent2/n2^3"]), type number),
8      #"Inserted Multiplication3" = Table.AddColumn(#"Calculate SE(RD)", "1.96*SE",
9      each [#"SE(RD)"] * 1.96, type number),
10     #"Calculate RD 2.5% CI" = Table.AddColumn(#"Inserted Multiplication3", "RD
11     2.5% CI", each [Risk difference] - [#"1.96*SE"], type number),
12     #"Calculate RD 97.5% CI" = Table.AddColumn(#"Calculate RD 2.5% CI", "RD
13     97.5% CI", each [Risk difference] + [#"1.96*SE"], type number),
14     #"Calculating events/1000 people" = Table.AddColumn(#"Calculate RD 97.5% CI",
15     "Events/1000", each [Risk difference] * 1000, type number),
16     #"Calculating 2.5% CI" = Table.AddColumn(#"Calculating events/1000 people",
17     "Events/1000 2.5% CI", each [#"RD 2.5% CI"] * 1000, type number),
18     #"Calculating 97.5% CI" = Table.AddColumn(#"Calculating 2.5% CI", "Events/1000
19     97.5% CI", each [#"RD 97.5% CI"] * 1000, type number),
20     #"Changed Type2" = Table.TransformColumnTypes(#"Calculating 97.5%
21     CI",{{"Events/1000 97.5% CI", Int64.Type}, {"Events/1000 2.5% CI", Int64.Type},
22     {"Events/1000", Int64.Type}}),
23     #"Removed Other Columns1" = Table.SelectColumns(#"Changed
24     Type2",{ "Analysis group", "Analysis number", "Analysis name", "Analysis group
25     name", "Outcome", "Experimental intervention", "Control intervention", "Effect
26     measure", "Number of events 1", "n1", "Number of events 2", "n2", "Mean", "CI start",
27     "CI end", "Sample size", "Events/1000", "Events/1000 2.5% CI", "Events/1000 97.5%
28     CI"}),
29     #"Subtracting CIs to calculate SE" = Table.AddColumn(#"Removed Other
30     Columns1", "SE", each [CI end] - [CI start], type number),
31     #"Dividing by 3.92 to calculate SE" = Table.TransformColumns(#"Subtracting CIs
32     to calculate SE", {{"SE", each _ / 3.92, type number}}),
33     #"Square Root of sample size" = Table.AddColumn(#"Dividing by 3.92 to calculate
34     SE", "Square Root(N)", each Number.Sqrt([Sample size]), type number),

```

```

1      #"Calculate SD" = Table.AddColumn("#Square Root of sample size", "SD", each
2      [#"Square Root(N)"] * [SE], type number),
3      #"SD squared" = Table.AddColumn("#Calculate SD", "SD^2", each
4      Number.Power([SD], 2), type number),
5      #"Choosing effect for MID precision calc" = Table.AddColumn("#SD squared",
6      "Events for MIDs", each if [Effect measure] = "Risk Ratio" then [Mean] else if [Effect
7      measure] = "Odds Ratio" then [Mean] else if [Effect measure] = "Risk Difference"
8      then [#"Events/1000"] else [Mean]),
9      #"Choosing 2.5% CI for MID precision calc" = Table.AddColumn("#Choosing effect
10     for MID precision calc", "2.5% CI for MIDs", each if [Effect measure] = "Risk Ratio"
11     then [CI start] else if [Effect measure] = "Odds Ratio" then [CI start] else if [Effect
12     measure] = "Risk Difference" then [#"Events/1000 2.5% CI"] else [CI start]),
13     #"Choosing 97.5% CI for MID precision calc" = Table.AddColumn("#Choosing
14     2.5% CI for MID precision calc", "97.5% CI for MIDs", each if [Effect measure] = "Risk
15     Ratio" then [CI end] else if [Effect measure] = "Odds Ratio" then [CI end] else if
16     [Effect measure] = "Risk Difference" then [#"Events/1000 97.5% CI"] else [CI end]),
17     #"Analyst inputting MIDs" = Table.AddColumn("#Choosing 97.5% CI for MID
18     precision calc", "MID", each if [Outcome] = "Person/participant generic health-related
19     quality of life - SF-36 physical component score" then 2 else if [Outcome] =
20     "Person/participant generic health-related quality of life - SF-36 mental component
21     score" then 3 else if [Effect measure] = "Risk Ratio" then 0.8 else if [Effect measure]
22     = "Hazard Ratio" then 0.8 else if [Effect measure] = "Odds Ratio" then 0.8 else null),
23     #"Calculating inverse MIDs (assuming linear scale) - for non-linear scale determine
24     differently" = Table.AddColumn("#Analyst inputting MIDs", "Inverse MID", each [MID]
25     * -1, type number),
26     #"Analyst inverse MID choice" = Table.AddColumn("#Calculating inverse MIDs
27     (assuming linear scale) - for non-linear scale determine differently", "Inverse MID
28     choice", each if [Effect measure] = "Risk Ratio" then 1.25 else if [Effect measure] =
29     "Hazard Ratio" then 1.25 else if [Effect measure] = "Odds Ratio" then 1.25 else
30     [Inverse MID]),
31     #"2.5% CI relative precision value determination" = Table.AddColumn("#Analyst
32     inverse MID choice", "2.5% CI MID relation", each if [MID] = null then null else if
33     [#"2.5% CI for MIDs"] >= [Inverse MID choice] then 1 else if [#"2.5% CI for MIDs"] >=
34     [MID] then 0 else if [#"2.5% CI for MIDs"] < [MID] then -1 else null),

```

```

1      #"97.5% CI relative precision value determination" = Table.AddColumn(#"2.5% CI
2 relative precision value determination", "97.5% CI MID relation", each if [MID] = null
3 then null else if [#"97.5% CI for MIDs"] < [Inverse MID choice] then -1 else if [#"97.5%
4 CI for MIDs"] < [MID] then 0 else if [#"97.5% CI for MIDs"] >= [MID] then 1 else null),
5      #"Changed Type1" = Table.TransformColumnTypes(#"97.5% CI relative precision
6 value determination",{{"2.5% CI MID relation", Int64.Type}, {"97.5% CI MID relation",
7 Int64.Type}}),
8      #"Combining precision CI values together" = Table.AddColumn(#"Changed
9 Type1", "Precision number", each Text.Combine({Text.From([#"2.5% CI MID
10 relation"], "en-GB"), Text.From([#"97.5% CI MID relation"], "en-GB")}, "="), type text),
11      #"Replaced Value1" = Table.ReplaceValue(#"Combining precision CI values
12 together","",null,Replacer.ReplaceValue,{"Precision number"}),
13      #"Imprecision MID calculation" = Table.AddColumn(#"Replaced Value1",
14 "Imprecision MID", each if [Precision number] = null then null else if [Precision
15 number] = "0=0" then "Not serious" else if [Precision number] = "1=1" then "Not
16 serious" else if [Precision number] = "-1=-1" then "Serious" else if [Precision number]
17 = "0=1" then "Serious" else if [Precision number] = "1=0" then "Serious" else if
18 [Precision number] = "-1=0" then "Serious" else if [Precision number] = "0=-1" then
19 "Not serious" else if [Precision number] = "-1=1" then "Very serious" else if [Precision
20 number] = "1=-1" then "Very serious" else "?"),
21      #"Merged Queries" = Table.NestedJoin(#"Imprecision MID calculation", {"Analysis
22 group", "Analysis number"}, #"OIS dichotomous", {"Analysis group", "Analysis
23 number"}, "OIS dichotomous", JoinKind.LeftOuter),
24      ///If there were continuous outcomes, you would also merge the query for OIS
25 continuous, get the OIS precision for that, merge that with the OIS precision for
26 dichotomous and then do the rest of the steps as usual
27      #"Expanded OIS dichotomous" = Table.ExpandTableColumn(#"Merged Queries",
28 "OIS dichotomous", {"OIS", "OIS Precision"}, {"OIS", "OIS Precision"}),
29      #"Replaced Value" = Table.ReplaceValue(#"Expanded OIS dichotomous","Not
30 applicable",null,Replacer.ReplaceValue,{"OIS Precision"}),
31      #"Analyst imprecision edit" = Table.AddColumn(#"Replaced Value", "Imprecision
32 analyst", each if [Outcome] = "Acute myocardial infarction" then "Very serious" else
33 null),

```

```

1      #"Final imprecision calculation" = Table.AddColumn(#"Analyst imprecision edit",
2      "Imprecision", each if [Imprecision analyst] <> null then [Imprecision analyst] else if
3      [Imprecision MID] <> null then [Imprecision MID] else if [OIS Precision] <> null then
4      [OIS Precision] else null),
5      #"Method used to determine precision" = Table.AddColumn(#"Final imprecision
6      calculation", "Method used", each if [Imprecision analyst] <> null then "Imprecision
7      analyst" else if [Imprecision MID] <> null then "Imprecision MID" else if [OIS
8      Precision] <> null then "OIS Precision" else null),
9      #"Removing not serious" = Table.AddColumn(#"Method used to determine
10     precision", "Imprecision for reason", each if [Imprecision] = "Not serious" then null
11     else [Imprecision]),
12     #"Removing method for not serious" = Table.AddColumn(#"Removing not
13     serious", "Method for reason", each if [Imprecision for reason] = null then null else
14     [Method used]),
15     #"Combining method and imprecision" = Table.AddColumn(#"Removing method
16     for not serious", "Method-Imprecision", each Text.Combine({[Method for reason],
17     [Imprecision for reason]}, "-"), type text),
18     #"Replaced Value2" = Table.ReplaceValue(#"Combining method and
19     imprecision", "", null, Replacer.ReplaceValue, {"Method-Imprecision"}),
20     #"Imprecision reason generation" = Table.AddColumn(#"Replaced Value2",
21     "Imprecision reason", each if [Outcome] = "Mortality" then "Downgraded by 1 interval
22     as agreed with committee" else if [#"Method-Imprecision"] = "Imprecision MID-
23     Serious" then "Serious imprecision because 95% CI crosses 1 MID (RR 0.8-1.25)"
24     else if [#"Method-Imprecision"] = "Imprecision MID-Very serious" then "Very serious
25     imprecision because 95% CI crosses 2 MIDs (RR 0.8-1.25)" else if [#"Method-
26     Imprecision"] = "OIS Precision-Serious" then "Serious imprecision because the
27     number of participants is between 30% and 100% of the optimal information size
28     (OIS)" else if [#"Method-Imprecision"] = "OIS Precision-Very serious" then "Very
29     serious imprecision because the number of participants is less than 30% of the
30     optimal information size (OIS) or the ratio between the confidence intervals exceed
31     2.5 (odds ratio)/3 (risk ratio) or both" else null),
32     #"Removed Other Columns2" = Table.SelectColumns(#"Imprecision reason
33     generation", {"Analysis group", "Analysis number", "Analysis name", "Number of
34     events 1", "n1", "Number of events 2", "n2", "Sample size", "Events/1000",

```

```

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

```

9 Number of studies

```

10 let
11   Source = Csv.Document(Web.Contents("source"),[Delimiter=","]),
12   #"Promoted Headers" = Table.PromoteHeaders(Source,
13 [PromoteAllScalars=true]),
14   #"Changed Type" = Table.TransformColumnTypes(#"Promoted
15 Headers",{{"Analysis group", Int64.Type}, {"Analysis number", Int64.Type}, {"Analysis
16 name", type text}, {"Subgroup", type any}, {"Applicability", type text}, {"Study", type
17 text}, {"Study year", Int64.Type}, {"GIV Mean", type number}, {"GIV SE", type
18 number}, {"Experimental mean", type number}, {"Experimental SD", type number},
19 {"Experimental cases", Int64.Type}, {"Experimental N", Int64.Type}, {"Control mean",
20 type number}, {"Control SD", type number}, {"Control cases", Int64.Type}, {"Control
21 N", Int64.Type}, {"O-E", type any}, {"Variance", type any}, {"Weight", type number},
22 {"Mean", type number}, {"CI start", type number}, {"CI end", type number},
23 {"Footnotes", type any}}),
24   #"Splitting by "" at "" to separate follow up time" = Table.SplitColumn(#"Changed
25 Type", "Analysis name", Splitter.SplitTextByDelimiter(" at ", QuoteStyle.Csv),
26 {"Analysis name.1", "Analysis name.2"}),
27   #"Changed Type1" = Table.TransformColumnTypes(#"Splitting by "" at "" to
28 separate follow up time",{{"Analysis name.1", type text}, {"Analysis name.2", type
29 text}}),
30   #"Renamed Columns" = Table.RenameColumns(#"Changed Type1",{{"Analysis
31 name.2", "Follow up time"}}),

```

```

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

```

```

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

```

19 Inconsistency, units

```

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

```

```

1  {"Overall estimates", type logical}, {"Test for subgroup differences", type logical},
2  {"Prediction interval", type logical}, {"Swap event and non-event", type logical}, {"CI
3  method", type text}, {"CI/PI level", type number}, {"Experimental cases", Int64.Type},
4  {"Experimental N", Int64.Type}, {"Control cases", Int64.Type}, {"Control N",
5  Int64.Type}, {"Mean", type number}, {"CI start", type number}, {"CI end", type
6  number}, {"PI start", type any}, {"PI end", type any}, {"Heterogeneity Tau2", type any},
7  {"Tau2 CI start", type any}, {"Tau2 CI end", type any}, {"Heterogeneity Chi2", type
8  number}, {"Heterogeneity df", Int64.Type}, {"Heterogeneity P", type number},
9  {"Heterogeneity I2", type number}, {"Effect Z", type number}, {"Effect P", type
10 number}, {"Subgroup Chi2", type any}, {"Subgroup df", type any}, {"Subgroup P", type
11 any}, {"Subgroup I2", type any}, {"ID", type text})),
12   #"Removed Columns" = Table.RemoveColumns(#"Changed Type",{ "PI start", "PI
13   end", "Heterogeneity Tau2", "Tau2 CI start", "Tau2 CI end", "Heterogeneity Chi2",
14   "Heterogeneity df", "Heterogeneity P", "Effect Z", "Effect P", "Subgroup Chi2",
15   "Subgroup df", "Subgroup P", "ID", "CI method", "Test for subgroup differences",
16   "Prediction interval", "Swap event and non-event", "Tau2 CI", "Heterogeneity
17   estimator", "Experimental intervention", "Control intervention"})),
18   #"Replaced Value1" = Table.ReplaceValue(#"Removed
19   Columns",100,0,Replacer.ReplaceValue,{"Heterogeneity I2"}),
20   #"Inserted Merged Column" = Table.AddColumn(#"Replaced Value1", "GRADE
21   group", each Text.Combine({Text.From([Analysis group], "en-GB"),
22   Text.From([Analysis number], "en-GB")}, "-"), type text),
23   #"Merged Queries" = Table.NestedJoin(#"Inserted Merged Column", {"GRADE
24   group"}, #"Number of studies", {"GRADE group"}, "Number of studies",
25   JoinKind.LeftOuter),
26   #"Expanded Number of studies" = Table.ExpandTableColumn(#"Merged Queries",
27   "Number of studies", {"Number of studies"}, {"Number of studies"}),
28   #"Inconsistency calculation" = Table.AddColumn(#"Expanded Number of studies",
29   "Inconsistency calculation", each if [Number of studies] = 1 then "Serious - single
30   study" else if [#"Heterogeneity I2"] < 0 then "?" else if [#"Heterogeneity I2"] < 40 then
31   "Not serious" else if [#"Heterogeneity I2"] < 60 then "Serious" else if [#"Heterogeneity
32   I2"] < 80 then "Very serious" else if [#"Heterogeneity I2"] < 100 then "Very serious -
33   too heterogenous to meaningfully pool?" else "?"),

```



```

1      #"Analyst edit of inconsistency" = Table.AddColumn("#Inconsistency calculation",
2      "Inconsistency analyst check", each if [Analysis name] = "Volume of allogenic blood
3      transfused (mL) at end of study" then "Serious" else if [Analysis name] = "Acute
4      myocardial infarction at end of study" then "Serious" else if [Analysis name] = "Length
5      of stay (days) at end of study" then "Very serious" else if [Analysis name] = "Total
6      blood loss volume at end of study" then "Very serious" else null),
7      #"Final inconsistency calculation" = Table.AddColumn("#Analyst edit of
8      inconsistency", "Inconsistency", each if [Inconsistency analyst check] <> null then
9      [Inconsistency analyst check] else [Inconsistency calculation]),
10     #"Inconsistency reason" = Table.AddColumn("#Final inconsistency calculation",
11     "Inconsistency reason", each if [Number of studies] = 1 then "Inconsistency: Single
12     study- downgraded once for inconsistency, as single study outcomes may otherwise
13     receive favourable ratings for inconsistency by default" else if [#"Heterogeneity I2"] <
14     40 then null else if [#"Heterogeneity I2"] < 60 then "Inconsistency: Downgraded once.
15     Serious heterogeneity (I2 = 40 to 60%) unexplained by subgroup analysis. Random
16     effects analysis used" else if [#"Heterogeneity I2"] < 80 then "Inconsistency:
17     Downgraded twice. Very serious heterogeneity (serious I2 = >60%) unexplained by
18     subgroup analysis. Random effects analysis used" else if [#"Heterogeneity I2"] <=
19     100 then "Inconsistency: Downgraded twice. Very serious heterogeneity (serious I2 =
20     >60%) unexplained by subgroup analysis. Random effects analysis used" else if
21     [#"Heterogeneity I2"] > 100 then "?" else if [Analysis name] = "Mortality" then null else
22     null),
23     #"Removed Columns1" = Table.RemoveColumns("#Inconsistency
24     reason",{"Inconsistency calculation", "Inconsistency analyst check"}),
25     #"Analyst editable units column" = Table.AddColumn("#Removed Columns1",
26     "Units", each if [Unit of effect measure] <> null then [Unit of effect measure] else if
27     [Outcome] = "Length of stay" then "days" else if [Outcome] = "Number of units of
28     allogenic blood transfused" then "units" else if [Outcome] = "Volume of allogenic
29     blood transfused" then "mL" else "Not applicable"),
30     #"Added Custom1" = Table.AddColumn("#Analyst editable units column", "|", each
31     "|"),
32     #"Analyst editable study type" = Table.AddColumn("#Added Custom1", "Study
33     type", each if Text.Contains([Analysis name], "non-randomised") then "Non-

```

```

1  randomised studies" else if Text.Contains([Analysis name], "randomised") then
2  "Randomised trials" else "Randomised trials"),
3      #"Analyst editable ROBINS-I used?" = Table.AddColumn(#"Analyst editable study
4  type", "ROBINS-I used?", each if [Study type] = "Randomised trials" then false else if
5  [Outcome] = "a" then true else if [Outcome] = "b" then false else true),
6      #"Analyst editable publication bias" = Table.AddColumn(#"Analyst editable
7  ROBINS-I used?", "Publication bias", each if [Outcome] = "a" then "Undetected" else
8  if [Outcome] = "b" then "Strongly suspected" else "Undetected"),
9      #"Analyst editable large effect" = Table.AddColumn(#"Analyst editable publication
10 bias", "Large effect", each if [Study type] = "Randomised trials" then "No" else if
11 [Outcome] = "a" then "No" else if [Outcome] = "b" then "Large" else if [Outcome] = "c"
12 then "Very large" else "No"),
13      #"Analyst editable plausible confounding" = Table.AddColumn(#"Analyst editable
14 large effect", "Plausible confounding", each if [Study type] = "Randomised trials" then
15 "No" else if [Outcome] = "a" then "No" else if [Outcome] = "b" then "Would reduce
16 demonstrated effect" else if [Outcome] = "c" then "Would suggest spurious effect"
17 else "No"),
18      #"Analyst editable dose response gradient" = Table.AddColumn(#"Analyst editable
19 plausible confounding", "Dose response gradient", each if [Study type] =
20 "Randomised trials" then "No" else if [Outcome] = "a" then "No" else if [Outcome] =
21 "b" then "No" else "No"),
22      #"Making other considerations column" = Table.AddColumn(#"Analyst editable
23 dose response gradient", "Other considerations", each Text.Combine({[Publication
24 bias], [Large effect], [Plausible confounding], [Dose response gradient]}, "
25 "), type text),
26      #"Replaced Value" = Table.ReplaceValue(#"Making other considerations
27 column", "Undetected
28 No
29 No
30 No", "None", Replacer.ReplaceText, {"Other considerations"}),
31      #"Analyst editable importance" = Table.AddColumn(#"Replaced Value",
32 "Importance", each if [Outcome] = "Health-related quality of life" then "Critical" else if
33 [Outcome] = "Mortality" then "Critical" else if [Outcome] = "a" then "Critical" else if
34 [Outcome] = "b" then "Important" else "Critical"),

```

```

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

```

8 Median baseline rate

```

9 let
10     Source = Csv.Document(Web.Contents("source"),[Delimiter=","]),
11     #"Promoted Headers" = Table.PromoteHeaders(Source,
12 [PromoteAllScalars=true]),
13     #"Changed Type" = Table.TransformColumnTypes(#"Promoted
14 Headers",{{"Analysis group", Int64.Type}, {"Analysis number", Int64.Type}, {"Analysis
15 name", type text}, {"Experimental cases", Int64.Type}, {"Experimental N",
16 Int64.Type}, {"Control cases", Int64.Type}, {"Control N", Int64.Type}, {"Mean", type
17 number}, {"CI start", type number}, {"CI end", type number}}),
18     #"Splitting by "" at "" to separate follow up time" = Table.SplitColumn(#"Changed
19 Type", "Analysis name", Splitter.SplitTextByDelimiter(" at ", QuoteStyle.Csv),
20 {"Analysis name.1", "Analysis name.2"}),
21     #"Changed Type1" = Table.TransformColumnTypes(#"Splitting by "" at "" to
22 separate follow up time",{{"Analysis name.1", type text}, {"Analysis name.2", type
23 text}}),
24     #"Renamed Columns" = Table.RenameColumns(#"Changed Type1",{{"Analysis
25 name.2", "Follow up time"}}),
26     #"Splitting by "" (" to separate scale information" = Table.SplitColumn(#"Renamed
27 Columns", "Analysis name.1", Splitter.SplitTextByDelimiter(" (", QuoteStyle.Csv),
28 {"Analysis name.1.1", "Analysis name.1.2"}),
29     #"Changed Type2" = Table.TransformColumnTypes(#"Splitting by "" (" to
30 separate scale information",{{"Analysis name.1.1", type text}, {"Analysis name.1.2",
31 type text}}),
32     #"Renamed Columns1" = Table.RenameColumns(#"Changed Type2",{{"Analysis
33 name.1.2", "Scale"}, {"Analysis name.1.1", "Protocol Outcome"}}),

```

```

1      #"Removing closing bracket" = Table.ReplaceValue(#"Renamed
2 Columns1",",",",",Replacer.ReplaceText,{"Scale"}),
3      #"Removed Duplicates" = Table.Distinct(#"Removing closing bracket"),
4      #"Inserted Division" = Table.AddColumn(#"Removed Duplicates", "Control group
5 rate", each [Control cases] / [Control N], type number),
6      #"Removed Columns" = Table.RemoveColumns(#"Inserted Division",{"Mean", "CI
7 start", "CI end", "Experimental cases", "Experimental N", "Control cases", "Control
8 N"}),
9      #"Duplicated Column" = Table.DuplicateColumn(#"Removed Columns", "Protocol
10 Outcome", "Protocol Outcome - Copy"),
11      #"Pivoted Column" = Table.Pivot(#"Duplicated Column", List.Distinct(#"Duplicated
12 Column"[#"Protocol Outcome - Copy"])", "Protocol Outcome - Copy", "Control group
13 rate", List.Average),
14 // Redo each time there is a new dataset
15      #"Combining median control events" =
16 Table.CombineColumns(Table.TransformColumnTypes(#"Pivoted Column", {"All-
17 cause mortality", type text}, {"Thromboembolic events after surgery", type text},
18 {"Pulmonary embolism", type text}, {"Deep vein thrombosis", type text}, {"Myocardial
19 infarction", type text}, {"Ischaemic stroke", type text}, {"Infection", type text}, {"All-
20 cause readmission", type text}, {"Seizures", type text}, {"Reoperation", type text}},
21 "en-GB"),{"All-cause mortality", "Thromboembolic events after surgery", "Pulmonary
22 embolism", "Deep vein thrombosis", "Myocardial infarction", "Ischaemic stroke",
23 "Infection", "All-cause readmission", "Seizures",
24 "Reoperation"},Combiner.CombineTextByDelimiter("", QuoteStyle.None),"Median
25 control group rate"),
26      #"Changed Type3" = Table.TransformColumnTypes(#"Combining median control
27 events",{"Median control group rate", type number})),
28      #"Producing GRADE number" = Table.AddColumn(#"Changed Type3", "GRADE
29 number", each Text.Combine({Text.From([Analysis group], "en-GB"),
30 Text.From([Analysis number], "en-GB")}, "-"), type text),
31      #"Filtered Rows" = Table.SelectRows(#"Producing GRADE number", each
32 ([Analysis group] <> 5 and [Analysis group] <> 6))
33 in
34      #"Filtered Rows"

```

1 **OIS dichotomous**

2 let

```

3   Source = Csv.Document(Web.Contents("source"),[Delimiter=","]),
4   #"Promoted Headers" = Table.PromoteHeaders(Source,
5   [PromoteAllScalars=true]),
6   #"Changed Type" = Table.TransformColumnTypes(#"Promoted
7   Headers",{{"Analysis group", Int64.Type}, {"Analysis number", Int64.Type}, {"Analysis
8   name", type text}, {"Analysis group name", type text}, {"Data source", type text},
9   {"Data source eligibility", type text}, {"Data type", type text}, {"Log-scale data", type
10  logical}, {"Outcome", type text}, {"Intervention grouping", type text}, {"Experimental
11  intervention", type any}, {"Control intervention", type any}, {"Subgroup by", type any},
12  {"Filter criteria", type any}, {"Experimental group label", type text}, {"Control group
13  label", type text}, {"Statistical method", type text}, {"Effect measure", type text}, {"Unit
14  of effect measure", type any}, {"Analysis model", type text}, {"Heterogeneity
15  estimator", type any}, {"Tau2 CI", type logical}, {"Subgroup estimates", type logical},
16  {"Overall estimates", type logical}, {"Test for subgroup differences", type logical},
17  {"Prediction interval", type logical}, {"Swap event and non-event", type logical}, {"CI
18  method", type text}, {"CI/PI level", type number}, {"Experimental cases", Int64.Type},
19  {"Experimental N", Int64.Type}, {"Control cases", Int64.Type}, {"Control N",
20  Int64.Type}, {"Mean", type number}, {"CI start", type number}, {"CI end", type
21  number}, {"PI start", type any}, {"PI end", type any}, {"Heterogeneity Tau2", type any},
22  {"Tau2 CI start", type any}, {"Tau2 CI end", type any}, {"Heterogeneity Chi2", type
23  number}, {"Heterogeneity df", Int64.Type}, {"Heterogeneity P", type number},
24  {"Heterogeneity I2", type number}, {"Effect Z", type number}, {"Effect P", type
25  number}, {"Subgroup Chi2", type any}, {"Subgroup df", type any}, {"Subgroup P", type
26  any}, {"Subgroup I2", type any}, {"ID", type text}}),
27   #"Removed Other Columns" = Table.SelectColumns(#"Changed Type",{ "Analysis
28   group", "Analysis number", "Analysis name", "Analysis group name", "Outcome",
29   "Intervention grouping", "Experimental intervention", "Control intervention",
30   "Experimental group label", "Control group label", "Effect measure", "Experimental
31   cases", "Experimental N", "Control cases", "Control N", "Mean", "CI start", "CI end",
32   "ID"}),

```

```

1      #"Removing mean difference and std. mean difference" =
2      Table.SelectRows(#"Removed Other Columns", each [Effect measure] <> "Mean
3      Difference" and [Effect measure] <> "Std. Mean Difference"),
4      #"Reordered Columns" = Table.ReorderColumns(#"Removing mean difference
5      and std. mean difference",{ "Analysis group", "Analysis number", "ID", "Analysis
6      name", "Analysis group name", "Outcome", "Intervention grouping", "Experimental
7      intervention", "Control intervention", "Experimental group label", "Control group label",
8      "Effect measure", "Experimental cases", "Experimental N", "Control cases", "Control
9      N"}),
10     #"Calculate proportion experimental group" = Table.AddColumn(#"Reordered
11     Columns", "p1", each [Experimental cases] / [Experimental N], type number),
12     #"Calculate proportion control group" = Table.AddColumn(#"Calculate proportion
13     experimental group", "p2", each [Control cases] / [Control N], type number),
14     #"Proportion unexposed with outcomes for GIV" = Table.AddColumn(#"Calculate
15     proportion control group", "prop unexposed", each 0.5),
16     #"Risk ratios only GIV" = Table.AddColumn(#"Proportion unexposed with
17     outcomes for GIV", "Risk ratio", each if [Effect measure] = "Risk Ratio" then [Mean]
18     else null),
19     #"Odds ratios only GIV" = Table.AddColumn(#"Risk ratios only GIV", "Odds ratio",
20     each if [Effect measure] = "Odds Ratio" then [Mean] else null),
21     #"Risk differences only GIV" = Table.AddColumn(#"Odds ratios only GIV", "Risk
22     difference", each if [Effect measure] = "Risk difference" then [Mean] else null),
23     #"Changed Type2" = Table.TransformColumnTypes(#"Risk differences only
24     GIV",{{"prop unexposed", type number}, {"Risk ratio", type number}, {"Odds ratio",
25     type number}, {"Risk difference", type number}}),
26     #"OR-1" = Table.AddColumn(#"Changed Type2", "OR-1", each [Odds ratio] - 1,
27     type number),
28     #"proportion unexposed+1" = Table.AddColumn(#"OR-1", "1+pe", each [prop
29     unexposed] + 1, type number),
30     #"Top equation portion" = Table.AddColumn(#"proportion unexposed+1",
31     "OR*prop unexposed", each [prop unexposed] * [Odds ratio], type number),
32     #"Bottom equation portion" = Table.AddColumn(#"Top equation portion", "(OR-
33     1)*(1+pe)", each [#"1+pe"] * [#"OR-1"], type number),

```

```

1      #"Odds ratio proportion exposed GIV" = Table.AddColumn(#"Bottom equation
2      portion", "OR prop exposed", each [#"OR*prop unexposed"] / [#"(OR-1)*(1+pe)"],
3      type number),
4      #"Risk ratio proportion exposed GIV" = Table.AddColumn(#"Odds ratio proportion
5      exposed GIV", "RR prop exposed", each [prop unexposed] * [Risk ratio], type
6      number),
7      #"Risk difference proportion exposed GIV" = Table.AddColumn(#"Risk ratio
8      proportion exposed GIV", "RD prop exposed", each [Risk difference] + [prop
9      unexposed], type number),
10     #"Added Conditional Column" = Table.AddColumn(#"Risk difference proportion
11     exposed GIV", "Proportion exposed", each if [Effect measure] = "Risk Ratio" then [RR
12     prop exposed] else if [Effect measure] = "Odds Ratio" then [OR prop exposed] else if
13     [Effect measure] = "Risk Difference" then [RD prop exposed] else null),
14     #"Renamed Columns" = Table.RenameColumns(#"Added Conditional
15     Column",{{"p1", "p1 pre"}, {"p2", "p2 pre"}}),
16     #"Added Conditional Column1" = Table.AddColumn(#"Renamed Columns", "p1",
17     each if [p1 pre] = null then [Proportion exposed] else [p1 pre]),
18     #"Added Conditional Column2" = Table.AddColumn(#"Added Conditional
19     Column1", "p2", each if [p2 pre] = null then [prop unexposed] else [p2 pre]),
20     #"Adding proportions" = Table.AddColumn(#"Added Conditional Column2", "pb",
21     each [p1] + [p2], type number),
22     #"Divide added proportions by 2" = Table.TransformColumns(#"Adding
23     proportions", {{"pb", each _ / 2, type number}}),
24     #"1" = Table.AddColumn(#"Divide added proportions by 2", "1", each 1),
25     #"1-pb" = Table.AddColumn(#"1", "qb", each [1] - [pb], type number),
26     #"1-p1" = Table.AddColumn(#"1-pb", "q1", each [1] - [p1], type number),
27     #"1-p2" = Table.AddColumn(#"1-p1", "q2", each [1] - [p2], type number),
28     #"p1-p2" = Table.AddColumn(#"1-p2", "Delta", each [p1] - [p2], type number),
29     #"Absolute only (delta)" = Table.TransformColumns(#"p1-p2",{{"Delta",
30     Number.Abs, type number}}),
31     #"za (assuming power 0.8)" = Table.AddColumn(#"Absolute only (delta)", "za",
32     each 0.841621234),
33     #"zb (assuming 2 tailed test, alpha 0.05)" = Table.AddColumn(#"za (assuming
34     power 0.8)", "zb", each 1.959963985),

```

```

1      #"2" = Table.AddColumn("#zb (assuming 2 tailed test, alpha 0.05)", "2", each 2),
2      #"2pq" = Table.AddColumn("#2", "sqrt2pq", each List.Product({[pb], [qb], [2]}), type
3      number),
4      #"sqrt(2pq)" = Table.TransformColumns("#2pq",{"sqrt2pq", Number.Sqrt, type
5      number})),
6      p1q1 = Table.AddColumn("#sqrt(2pq)", "p1q1", each [p1] * [q1], type number),
7      p2q2 = Table.AddColumn(p1q1, "p2q2", each [p2] * [q2], type number),
8      sumpqs = Table.AddColumn(p2q2, "sqrtsumpqs", each [p1q1] + [p2q2], type
9      number),
10     #"sqrt(sumpqs)" = Table.TransformColumns(sumpqs,{"sqrtsumpqs",
11     Number.Sqrt, type number})),
12     #"sqrt2pq * za" = Table.AddColumn("#sqrt(sumpqs)", "sqrt2pq*za", each [sqrt2pq]
13     * [za], type number),
14     #"sqrtsumpqs * zb" = Table.AddColumn("#sqrt2pq * za", "sqrtsumpqs*zb", each
15     [sqrtsumpqs] * [zb], type number),
16     #"Adding the two to make the top" = Table.AddColumn("#sqrtsumpqs * zb", "Top",
17     each [#"sqrt2pq*za"] + [#"sqrtsumpqs*zb"], type number),
18     #"Dividing the top by delta" = Table.AddColumn("#Adding the two to make the
19     top", "sss", each [Top] / [Delta], type number),
20     #"Squaring the value" = Table.AddColumn("#Dividing the top by delta", "OIS",
21     each Number.Power([sss], 2), type number),
22     #"Rounding it up to make the OIS" = Table.TransformColumns("#Squaring the
23     value",{"OIS", each Number.Round(_, 0), type number})),
24     #"Removed Other Columns1" = Table.SelectColumns("#Rounding it up to make
25     the OIS",{"Analysis group", "Analysis number", "Analysis name", "Analysis group
26     name", "Outcome", "Intervention grouping", "Experimental intervention", "Control
27     intervention", "Experimental group label", "Control group label", "Effect measure",
28     "Experimental cases", "Experimental N", "Control cases", "CI start", "CI end", "Control
29     N", "OIS"}),
30     #"Inserted Addition" = Table.AddColumn("#Removed Other Columns1", "Total N",
31     each [Experimental N] + [Control N], Int64.Type),
32     #"Inserted Multiplication" = Table.AddColumn("#Inserted Addition", "OIS two
33     arms", each [OIS] * 2, type number),

```



```

1      #"OIS proportion calculation" = Table.AddColumn("#Inserted Multiplication", "Prop
2 OIS", each [Total N] / [OIS two arms], type number),
3      #"CI proportion calculation" = Table.AddColumn("#OIS proportion calculation",
4 "Prop CI", each [CI end] / [CI start], type number),
5      #"RR CI proportion conditional" = Table.AddColumn("#CI proportion calculation",
6 "RR Prop CI", each if [Effect measure] = "Risk Ratio" then [Prop CI] else null),
7      #"OR CI proportion conditional" = Table.AddColumn("#RR CI proportion
8 conditional", "OR Prop CI", each if [Effect measure] = "Odds Ratio" then [Prop CI]
9 else null),
10     #"For RR and OR proportions - replace null with 0" = Table.ReplaceValue("#OR CI
11 proportion conditional",null,0,Replacer.ReplaceValue,{"RR Prop CI", "OR Prop CI"}),
12     #"Changed Type1" = Table.TransformColumnTypes("#For RR and OR proportions
13 - replace null with 0",{"RR Prop CI", type number}, {"OR Prop CI", type number}),
14     #"Precision conditional determination" = Table.AddColumn("#Changed Type1",
15 "Precision", each if [Effect measure] = "Hazard Ratio" then "Cannot be determined
16 using this method" else if [RR Prop CI] >= 3 then "Very serious concerns regarding
17 optimal information size" else if [OR Prop CI] >= 2.5 then "Very serious concerns
18 regarding optimal information size" else if [Prop OIS] >= 1 then "No concerns
19 regarding optimal information size" else if [Prop OIS] >= 0.3 then "Serious concerns
20 regarding optimal information size" else if [Prop OIS] < 0.3 then "Very serious
21 concerns regarding optimal information size" else "?"),
22     Custom1 = Table.AddColumn("#Precision conditional determination", "OIS
23 Precision", each if [Effect measure] = "Hazard Ratio" then "Not applicable" else if [RR
24 Prop CI] >= 3 then "Very serious" else if [OR Prop CI] >= 2.5 then "Very serious" else
25 if [Prop OIS] >= 1 then "Not serious" else if [Prop OIS] >= 0.3 then "Serious" else if
26 [Prop OIS] < 0.3 then "Very serious" else "Not serious")
27 in
28     Custom1

```

29 **GRADE table**

```

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

```

```

1      #"Changed Type" = Table.TransformColumnTypes(#"Promoted
2 Headers",{{"Analysis group", Int64.Type}, {"Analysis number", Int64.Type}, {"Analysis
3 name", type text}, {"Subgroup", type any}, {"Applicability", type text}, {"Study", type
4 text}, {"Study year", Int64.Type}, {"GIV Mean", type number}, {"GIV SE", type
5 number}, {"Experimental mean", type number}, {"Experimental SD", type number},
6 {"Experimental cases", Int64.Type}, {"Experimental N", Int64.Type}, {"Control mean",
7 type number}, {"Control SD", type number}, {"Control cases", Int64.Type}, {"Control
8 N", Int64.Type}, {"O-E", type any}, {"Variance", type any}, {"Weight", type number},
9 {"Mean", type number}, {"CI start", type number}, {"CI end", type number},
10 {"Footnotes", type any}}),
11      #"Splitting by "" at "" to separate follow up time" = Table.SplitColumn(#"Changed
12 Type", "Analysis name", Splitter.SplitTextByDelimiter(" at ", QuoteStyle.Csv),
13 {"Analysis name.1", "Analysis name.2"}),
14      #"Changed Type1" = Table.TransformColumnTypes(#"Splitting by "" at "" to
15 separate follow up time",{{"Analysis name.1", type text}, {"Analysis name.2", type
16 text}}),
17      #"Renamed Columns" = Table.RenameColumns(#"Changed Type1",{{"Analysis
18 name.2", "Follow up time"}}),
19      #"Splitting by "" ("" to separate scale information" = Table.SplitColumn(#"Renamed
20 Columns", "Analysis name.1", Splitter.SplitTextByDelimiter(" (", QuoteStyle.Csv),
21 {"Analysis name.1.1", "Analysis name.1.2"}),
22      #"Changed Type2" = Table.TransformColumnTypes(#"Splitting by "" ("" to
23 separate scale information",{{"Analysis name.1.1", type text}, {"Analysis name.1.2",
24 type text}}),
25      #"Renamed Columns1" = Table.RenameColumns(#"Changed Type2",{{"Analysis
26 name.1.2", "Scale"}, {"Analysis name.1.1", "Protocol Outcome"}}),
27      #"Removing closing bracket" = Table.ReplaceValue(#"Renamed
28 Columns1",")","",Replacer.ReplaceText,{"Scale"}),
29      #"Making short outcome" = Table.AddColumn(#"Removing closing bracket",
30 "Outcome short", each if [Protocol Outcome] = "Thromboembolic events after
31 surgery" then "Thromboembolic events" else if [Protocol Outcome] = "All-cause
32 mortality" then "Mortality" else if [Protocol Outcome] = "Pulmonary embolism" then
33 "PE" else if [Protocol Outcome] = "Deep vein thrombosis" then "DVT" else if [Protocol
34 Outcome] = "Myocardial infarction" then "MI" else if [Protocol Outcome] = "Ischaemic

```

```

1  stroke" then "Stroke" else if [Protocol Outcome] = "Infection" then "Infection" else if
2  [Protocol Outcome] = "All-cause readmission" then "Readmission" else if [Protocol
3  Outcome] = "Readmission due to thrombosis" then "Thrombosis readmission" else
4  "?"),
5      #"Merged Queries" = Table.NestedJoin(#"Making short outcome", {"Study ID"},
6      #"Data rows", {"Study ID"}, "Critical appraisal no edits", JoinKind.LeftOuter),
7      #"Expanded Critical appraisal no edits" = Table.ExpandTableColumn(#"Merged
8      Queries", "Critical appraisal no edits", {"Short population", "Short comparison"},
9      {"Short population", "Short comparison"}),
10     #"Removed Duplicates" = Table.Distinct(#"Expanded Critical appraisal no edits"),
11     #"Removed Other Columns1" = Table.SelectColumns(#"Removed
12     Duplicates",{"Analysis group", "Analysis number", "Protocol Outcome", "Scale",
13     "Follow up time", "Applicability", "Study", "Study ID"}),
14     #"Duplicated Column" = Table.DuplicateColumn(#"Removed Other Columns1",
15     "Protocol Outcome", "Protocol Outcome - Copy"),
16     #"Merged Queries5" = Table.NestedJoin(#"Duplicated Column", {"Study ID"},
17     #"Study level data", {"Study ID"}, "Timepoint", JoinKind.LeftOuter),
18     #"Expanded Timepoint" = Table.ExpandTableColumn(#"Merged Queries5",
19     "Timepoint", {"Follow-up"}, {"Follow-up"}),
20     #"Renamed Columns3" = Table.RenameColumns(#"Expanded
21     Timepoint",{"Follow-up", "Average timepoint (days)"}),
22     #"Removed Columns6" = Table.RemoveColumns(#"Renamed Columns3",{"Study
23     ID", "Study"}),
24     #"Pivoted Column" = Table.Pivot(#"Removed Columns6", List.Distinct(#"Removed
25     Columns6",{"Protocol Outcome - Copy"}), "Protocol Outcome - Copy", "Average
26     timepoint (days)", List.Average),
27     //Redo this sum with each new dataset
28     #"Inserted Sum" = Table.AddColumn(#"Pivoted Column", "Average timepoint
29     (days)", each List.Sum({["All-cause mortality"], [Thromboembolic events after
30     surgery], [Pulmonary embolism], [Deep vein thrombosis], [Myocardial infarction],
31     [Ischaemic stroke], [Infection], ["All-cause readmission"], [Seizures], [Reoperation]})),
32     type number),
33     #"Changed Type4" = Table.TransformColumnTypes(#"Inserted Sum",{"Average
34     timepoint (days)", Int64.Type}),

```

```

1      #"Inserted Merged Column" = Table.AddColumn(#"Changed Type4", "GRADE
2 group", each Text.Combine({Text.From([Analysis group], "en-GB"),
3 Text.From([Analysis number], "en-GB")}, "-"), type text),
4      #"Removed Other Columns" = Table.SelectColumns(#"Inserted Merged
5 Column",{ "Analysis group", "Analysis number", "Protocol Outcome", "Scale", "Follow
6 up time", "Applicability", "Average timepoint (days)", "GRADE group"}),
7      #"Reordered Columns" = Table.ReorderColumns(#"Removed Other
8 Columns",{ "Analysis group", "Analysis number", "GRADE group", "Protocol
9 Outcome", "Scale", "Follow up time", "Average timepoint (days)"}),
10     #"Filtered Rows" = Table.SelectRows(#"Reordered Columns", each ([Analysis
11 group] <> 5 and [Analysis group] <> 6)),
12     #"Merged Queries1" = Table.NestedJoin(#"Filtered Rows", {"GRADE group"},
13 #"Risk of bias", {"GRADE code"}, "Risk of bias", JoinKind.LeftOuter),
14     #"Expanded Risk of bias" = Table.ExpandTableColumn(#"Merged Queries1", "Risk
15 of bias", {"RoB", "RoB reason"}, {"RoB", "RoB reason"}),
16     #"Merged Queries3" = Table.NestedJoin(#"Expanded Risk of bias", {"GRADE
17 group"}, #"Inconsistency, units", {"GRADE group"}, "Inconsistency",
18 JoinKind.LeftOuter),
19     #"Expanded Inconsistency" = Table.ExpandTableColumn(#"Merged Queries3",
20 "Inconsistency", {"Importance", "Inconsistency", "Inconsistency reason", "Other
21 considerations", "ROBINS-I used?", "Study type", "Units"}, {"Importance",
22 "Inconsistency.1", "Inconsistency reason", "Other considerations", "ROBINS-I used?",
23 "Study type", "Units"}),
24     #"Renamed Columns2" = Table.RenameColumns(#"Expanded
25 Inconsistency",{"Inconsistency.1", "Inconsistency"}),
26     #"Merged Queries2" = Table.NestedJoin(#"Renamed Columns2", {"GRADE
27 group"}, Indirectness, {"GRADE code"}, "Indirectness", JoinKind.LeftOuter),
28     #"Expanded Indirectness" = Table.ExpandTableColumn(#"Merged Queries2",
29 "Indirectness", {"Indirectness", "Indirectness reason"}, {"Indirectness", "Indirectness
30 reason"}),
31     #"Merged Queries4" = Table.NestedJoin(#"Expanded Indirectness", {"GRADE
32 group"}, Imprecision, {"GRADE group"}, "Imprecision", JoinKind.LeftOuter),

```

```

1   #"Expanded Imprecision" = Table.ExpandTableColumn(#"Merged Queries4",
2   "Imprecision", {"Imprecision", "Imprecision reason", "Sample size"}, {"Imprecision",
3   "Imprecision reason", "Sample size"}),
4   #"Merged Queries6" = Table.NestedJoin(#"Expanded Imprecision", {"GRADE
5   group"}, #"Number of studies", {"GRADE group"}, "Number of studies",
6   JoinKind.LeftOuter),
7   #"Expanded Number of studies" = Table.ExpandTableColumn(#"Merged
8   Queries6", "Number of studies", {"Number of studies"}, {"Number of studies.1"}),
9   #"Added Conditional Column1" = Table.AddColumn(#"Expanded Number of
10  studies", "ROBINS-I adjustment", each if [Study type] = "Randomised trials" then 0
11  else if [#"ROBINS-I used?"] = true then 0 else 2),
12  #"Calculating RoB relative number" = Table.AddColumn(#"Added Conditional
13  Column1", "RoB number", each if [RoB] = "Not serious" then 0 else if [RoB] =
14  "Serious" then 1 else if [RoB] = "Very serious" then 2 else "?"),
15  #"Calculating indirectness relative number" = Table.AddColumn(#"Calculating RoB
16  relative number", "Indirectness number", each if [Indirectness] = "Not serious" then 0
17  else if [Indirectness] = "Serious" then 1 else if [Indirectness] = "Very serious" then 2
18  else "?"),
19  #"Calculating inconsistency number" = Table.AddColumn(#"Calculating
20  indirectness relative number", "Inconsistency number", each if
21  Text.StartsWith([Inconsistency], "Not serious") then 0 else if
22  Text.StartsWith([Inconsistency], "Serious") then 1 else if
23  Text.StartsWith([Inconsistency], "Very serious") then 2 else "?"),
24  #"Calculating imprecision relative number" = Table.AddColumn(#"Calculating
25  inconsistency number", "Imprecision number", each if [Imprecision] = "Not serious"
26  then 0 else if [Imprecision] = "Serious" then 1 else if [Imprecision] = "Very serious"
27  then 2 else "?"),
28  #"Changed Type3" = Table.TransformColumnTypes(#"Calculating imprecision
29  relative number",{"RoB number", Int64.Type}, {"Indirectness number", Int64.Type},
30  {"Inconsistency number", Int64.Type}, {"Imprecision number", Int64.Type},
31  {"ROBINS-I adjustment", Int64.Type}},
32  #"Inserted Sum1" = Table.AddColumn(#"Changed Type3", "GRADE number",
33  each List.Sum({"ROBINS-I adjustment", [RoB number], [Indirectness number],
34  [Inconsistency number], [Imprecision number]}), Int64.Type),

```

```

1      #"Added Conditional Column" = Table.AddColumn(#"Inserted Sum1", "GRADE",
2      each if [GRADE number] = 0 then "High" else if [GRADE number] = 1 then
3      "Moderate" else if [GRADE number] = 2 then "Low" else if [GRADE number] >= 3
4      then "Very low" else "?"),
5      #"Inserted Merged Column1" = Table.AddColumn(#"Added Conditional Column",
6      "Full reasons", each Text.Combine({[RoB reason], [Inconsistency reason],
7      [Indirectness reason], [Imprecision reason]}), "
8      "), type text),
9      #"Completing reasons" = Table.AddColumn(#"Inserted Merged Column1",
10     "Reasons", each if [Full reasons] = "" then "No downgrading required" else [Full
11     reasons]),
12     #"Removed Columns5" = Table.RemoveColumns(#"Completing reasons",{"Full
13     reasons"}),
14     #"Analyst edits for scale column" = Table.AddColumn(#"Removed Columns5",
15     "Outcome scale", each if [Scale] = "days" then "Not applicable" else if [Scale] = "mL"
16     then "Not applicable" else if [Scale] = "units" then "Not applicable" else if [Scale] <>
17     null then [Scale] else if [Protocol Outcome] = "Mortality" then "Not applicable" else
18     "Not applicable"),
19     #"Risk of bias short reason" = Table.AddColumn(#"Analyst edits for scale column",
20     "Short RoB reason", each if [RoB reason] <> null then "Risk of Bias" else null),
21     #"Inconsistency short reason" = Table.AddColumn(#"Risk of bias short reason",
22     "Inconsistency short reason", each if [Inconsistency reason] <> null then
23     "Inconsistency" else null),
24     #"Indirectness short reason" = Table.AddColumn(#"Inconsistency short reason",
25     "Indirectness short reason", each if [Indirectness reason] <> null then "Indirectness"
26     else null),
27     #"Imprecision short reason" = Table.AddColumn(#"Indirectness short reason",
28     "Imprecision short reason", each if [Imprecision reason] <> null then "Imprecision"
29     else null),
30     #"Combining short reasons" = Table.AddColumn(#"Imprecision short reason",
31     "Short reason", each Text.Combine({[Short RoB reason], [Inconsistency short
32     reason], [Indirectness short reason], [Imprecision short reason]}, ", ", type text),

```

```

1      #"Removed Columns2" = Table.RemoveColumns(#"Combining short
2      reasons",{ "Short RoB reason", "Inconsistency short reason", "Indirectness short
3      reason", "Imprecision short reason"}),
4      #"Added Custom" = Table.AddColumn(#"Removed Columns2", "Scale: ", each
5      "Scale: "),
6      #"Added Custom1" = Table.AddColumn(#"Added Custom", "Units: ", each "Units:
7      "),
8      #"Added Custom2" = Table.AddColumn(#"Added Custom1", "Study types:", each
9      "Study types: "),
10     #"Inserted Merged Column2" = Table.AddColumn(#"Added Custom2", "Scale:
11     value", each Text.Combine({{"Scale: "}, [Outcome scale]}, ""), type text),
12     #"Inserted Merged Column3" = Table.AddColumn(#"Inserted Merged Column2",
13     "Units: value", each Text.Combine({{"Units: "}, [Units]}, ""), type text),
14     #"Inserted Merged Column5" = Table.AddColumn(#"Inserted Merged Column3",
15     "Study type: value", each Text.Combine({{"Study types:"}, [Study type]}, ""), type
16     text),
17     #"Removed Columns3" = Table.RemoveColumns(#"Inserted Merged
18     Column5",{ "Scale: ", "Units: ", "Study types:"}),
19     #"Inserted Merged Column4" = Table.AddColumn(#"Removed Columns3",
20     "Outcome table", each Text.Combine({[Protocol Outcome], [Follow up time]}, " at "),
21     type text),
22     #"Lowercased Text" = Table.TransformColumns(#"Inserted Merged
23     Column4",{{"Scale: value", Text.Lower, type text}, {"Units: value", Text.Lower, type
24     text}, {"Study type: value", Text.Lower, type text}}),
25     #"Replaced Value" = Table.ReplaceValue(#"Lowercased
26     Text","ml","mL",Replacer.ReplaceText,{"Units: value"}),
27     #"Merged Columns" = Table.CombineColumns(#"Replaced Value",{ "Outcome
28     table", "Study type: value", "Scale: value", "Units:
29     value"},Combiner.CombineTextByDelimiter(", ", QuoteStyle.None),"Outcome full
30     table"),
31     #"Added Custom3" = Table.AddColumn(#"Merged Columns", "Risk of bias: ", each
32     "Risk of bias: "),
33     #"Added Custom4" = Table.AddColumn(#"Added Custom3", "Indirectness: ", each
34     "Indirectness: "),

```

```

1      # "Added Custom5" = Table.AddColumn(# "Added Custom4", "Inconsistency: ",
2  each "Inconsistency: "),
3      # "Added Custom6" = Table.AddColumn(# "Added Custom5", "Imprecision: ", each
4  "Imprecision: "),
5      # "Added Custom7" = Table.AddColumn(# "Added Custom6", "Other concerns: ",
6  each "Other considerations: "),
7      # "Inserted Merged Column6" = Table.AddColumn(# "Added Custom7", "Risk of
8  bias: value", each Text.Combine({[# "Risk of bias: "], [RoB]}, ""), type text),
9      # "Inserted Merged Column7" = Table.AddColumn(# "Inserted Merged Column6",
10 "Indirectness: value", each Text.Combine({[# "Indirectness: "], [Indirectness]}, ""), type
11 text),
12      # "Inserted Merged Column8" = Table.AddColumn(# "Inserted Merged Column7",
13 "Inconsistency: value", each Text.Combine({[# "Inconsistency: "], [Inconsistency]}, ""),
14 type text),
15      # "Inserted Merged Column9" = Table.AddColumn(# "Inserted Merged Column8",
16 "Imprecision: value", each Text.Combine({[# "Imprecision: "], [Imprecision]}, ""), type
17 text),
18      # "Inserted Merged Column10" = Table.AddColumn(# "Inserted Merged Column9",
19 "Other considerations: value", each Text.Combine({[# "Other concerns: "], [Other
20 considerations]}, ""), type text),
21      # "Inserted Merged Column11" = Table.AddColumn(# "Inserted Merged Column10",
22 "GRADE components", each Text.Combine({[# "Risk of bias: value"], [# "Indirectness:
23 value"], [# "Inconsistency: value"], [# "Imprecision: value"], [# "Other considerations:
24 value"]}, "
25 "), type text),
26      # "Removed Columns4" = Table.RemoveColumns(# "Inserted Merged
27 Column11", {"Risk of bias: ", "Indirectness: ", "Inconsistency: ", "Imprecision: ", "Other
28 concerns: ", "Risk of bias: value", "Indirectness: value", "Inconsistency: value",
29 "Imprecision: value", "Other considerations: value"}),
30      # "Filtered Rows1" = Table.SelectRows(# "Removed Columns4", each ([Scale] <>
31 "risk difference"))
32 in
33      # "Filtered Rows1"
34

```