Delft University of Technology

Calibration scores for discrete random variables

Hugo van der Poel

July 13, 2023

Calibration score for discrete random variables

by

Hugo van der Poel

to obtain the degree of Bachelor of Science
at Delft University of Technology

Student number: 5067553
Project duration: April 24, 2023 - July 14, 2023
Thesis committee: Dr. G. F. Nane, TU Delft, supervisor
Dr. A. F. F. Derumigny, TU Delft

Abstract

In this thesis, more research is done to investigate Hanea's and Nane's calibration score for discrete random variables. Also, slight adjustments of Hanea's and Nane's calibration score are introduced as well as investigating the behaviour of these calibration scores. For this, both real data as well as simulations are used to validate the procedure.
In section 2, an overview is given of how the validation of the quality of the assessments of the different experts for events which either occur or don't occur is given. In section 3, the focus will be on looking at different properties of this calibration score using cases where all events get assigned to the same bin. Also, in section 3, other calibration scores which are slight adjustments of Hanea's and Nane's calibration score are introduced as well as looking at the properties of these calibration scores using cases where all events get assigned to the same bin. In section 4, data consisting of experts answering calibration questions is used to further investigate the behaviour of the calibration scores. At the end of the report, the concluding remarks, references and appendices for the R code used during this project can be found.

Contents

1 Introduction **8**

2 Structured expert judgement **9**

2.1 Data to validate quality of assesment 9

2.1.1 Example of bin assignment 10

2.2 Validation of quality assessment of experts 11

2.2.1 Calibration score using chi-squared distribution 12

2.2.2 Calibration score using binomial distribution 13

3 Some properties of calibration scores for discrete random variables using simulated data **14**

3.1 All events assigned to same bin 14

3.2 Hanea's and Nane's calibration score using different formulas for p-value 17

3.3 Calibration score and sample size 20

3.4 Summarizing results 21

4 Some properties of calibration scores for discrete random variables with data **23**

4.1 DataSIPS 23

4.1.1 Behaviour of calibration scores 23

4.2 For dataACE_IDEA 26

4.2.1 Behaviour of calibration scores 26

4.3 For dataACE_GJP 28

4.3.1 Behaviour of calibration scores 28

4.4 Summarizing results 31

5 Concluding remarks **33**

6 Future research **35**

References **36**

A More tables used in section 2 **39**

B R code for section 3 **45**

C R code for section 4 **56**

D R code used in appendices A and B **75**

1 Introduction

When a government or company makes policy, it is often needed to elicit the probability whether events of interests will occur or not. To do so, these governments and companies often ask experts for their assessments. Before using the assessments, it is important to validate the quality of the assessments of the different experts. An example of an organization which uses this type of forecasting to serve governments, companies NGOS and nonprofits is The Good Judgement Project. This organization uses this forecasting for example to predict topics with regards to the USA election or topics with regards to the Russia - Ukraine war (Good Judgment Inc, 2023).

Along with the Brier score (Brier, 1950), other scoring rules have been developed in order to validate the quality of the assessment of the different experts. Inspired by the continuous setting, Cooke (1991) developed a calibration and information score. The robustness of the calibration score is however guaranteed only by using tens (or sometimes hundreds) of calibration questions (Cooke, Mendel and Thijs, 1988), which is, in practice, inefficient. The reason that this number of calibration questions is needed to guarantee the robustness of Cooke's calibration score is because Cooke's calibration score is based on an asymptotic distribution. Hanea and Nane (2019) developed a calibration score based on an exact rather than asymptotic distributional result, which decreases the number of calibration questions used but this calibration score sometimes gives inexplicably small values.

During this bachelorproject, more research is done to investigate the properties of the Hanea's and Nane's calibration score and to look at slight adjustment of Hanea's and Nane's calibration score. Both simulated data and real data will be used to investigate the behaviour of these calibration scores.

2 Structured expert judgement

When a government or company makes policy, it is often needed to elicit the probability whether events of interests will occur or not. To do so, these governments and companies often ask experts for their assessments. Before using the assessments, it is important to validate the quality of the assessments of the different experts. This section will give an overview of how this validation is done.

2.1 Data to validate quality of assesment

In order to collect the data which is used to validate the quality of the assesments of the different experts, calibration questions are formulated. A calibration question is a question where the answer is known (by the analyst but not the expert or unknown but known in the very near future) and is related to the questions of interest where the answer is not yet known. The answers the experts give on these calibration questions form the data, which are used to determine the quality of the assessments of the different experts.

There are different types of calibration questions. In this thesis only the type of question where the answers whether the event will occur or not is considered. Note that assessing the probability of occurrence for certain events of interest equates to eliciting bivariate random variables. Say X is a bivariate random variable such that P(X = 1) = p and P(X = 0) = 1 - p, where X = 1 means that the event occurs and X = 0 means that the event does not occur.

Experts are asked to give an estimation of p. Using this estimation of p for each expert, Cooke, Mendel and Thijs came up with a method where the event whose occurrence is modelled by X is assigned to a probability bin (Cooke, Mendel and Thijs). In this method, any positive integer greater than 2 can be used as the amount of bins. In (Hanea and Nane, 2019) and in my bachelor thesis, the focus will be on using ten bins, denoted as B_1, B_2, ..., B_{10}. If an event is assigned to a bin B_i with $i \in \{1, 2, ..., 10\}$, this means that the expert associates the probability of occurrence with p_i. An event gets for instance assigned to B_3 if the best estimation about the probability of occurrence p of that expert is anywhere between 0.2 and 0.3. In this method the middle value of this interval is chosen, which is 0.25. Therefore in this method, $p_1 = 0.05$, $p_2 = 0.15$, ..., $p_{10} = 0.95$ and let p = (p_1, p_2, ..., p_{10}).

Say we have n binary random variables X_1, X_2, ..., X_n that model the probability of occurrence of n events. Let n_i be the number of events which got assigned to bin B_i, with $i \in \{1, 2, ..., 10\}$. Clearly, n = $\sum_{i=1}^{10} n_i$, as the expert assigns each separate event to exactly one of the ten bins. Given the outcome of each event, we consider

$$s_i = \frac{\sum_{j=1}^{n} X_j \cdot \mathbb{1}_{\{X_j \in B_i\}}}{n_i},$$

for i = 1, 2, ..., 10, where $1_{\{X_j \in B_i\}} = 1$ if $X_j = 1$ if X_j is assigned to B_i and 0 otherwise. The vector s = (s_1, s_2, ..., s_{10}) is often called the empirical probability vector. Preferably, s_i is close to p_i for i = 1, 2, ..., 10 and so it follows that taking as null hypothesis $H_0 : s_i = p_i$ is a natural choice (Hanea and Nane, 2019).

2.1.1 Example of bin assignment

In subsection 2.1, an overview was given of how to collect the data to validate the quality of the assessment of the experts using the bin assignment method introduced in (Cooke, Mendel and Thijs). Here an example is given of how this bin assignment works. Say we have 20 binary random variables $X_1, X_2, ...X_{20}$, where $X_1, X_2, ...X_{20}$ could for instance be whether it will rain tomorrow, whether it will rain in two days, ..., whether it will rain in 20 days, respectively. Then each event gets assigned to exactly one bin. One could visualize this by seeing the events as balls and putting each ball in one of the ten bins, see figure 1.

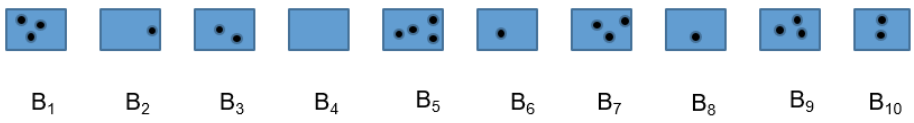

Figure 1: Vizualisation of bin assignment [1]

As can be seen in figure 1, three events got assigned to bin B_1, one event to bin B_2, two events to bin B_3, zero events to bin B_4, etc. Therefore, in this example, (n_1, n_2, n_3, n_4, n_5, n_6, n_7, n_8, n_9, n_{10}) = (3, 1, 2, 0, 4, 1, 3, 1, 3, 2).

[1] Figure 1 is taken from a slide from Hanea and Nane about their article (Hanea and Nane, 2019).

Suppose that 2 out of 3 events occur in bin B_7, then $s_7 = \frac{2}{3}$. If 0 out of 1 events occur in bin B_8, then $s_8 = 0$. So, as was stated mathematically in subsection 2.1, s_i is how many events in bin B_i occur, divided by how many events are in bin B_i.
Once the outcomes of the events in this example are known, we'd like to know how good the expert answered the calibration questions. The closer s_i is to p_i for i = 1, 2, ..., 10, the better the expert performed. That is, the closer s_1 is to 0.05, the closer s_2 is to 0.15, ..., the closer s_{10} is to 0.95, the better the expert performed. Given the bin assignment in this example after the expert answered the calibration questions, it would be best if 0 out of 3 events occurred in bin B_1, as zero is closer to 0.05 than $\frac{1}{3}$. Similarly, it would be best if 0 out of 1 events happened in bin B_2, as zero is closer to 0.15 than 1. Using similar reasoning, it would be best if 2 out of 4 events occurred in bin B_5 and 2 out of 2 events occurred in bin B_{10}.

2.2 Validation of quality assessment of experts

In subsection 2.1, an overview is given of how to collect the data to validate the quality of assessment of the different experts. This data is then used to evaluate the assessments. In (Morgan et Al. ,1979) four criteria for evaluating probability assessments are discussed (these criteria are attributed to Sarah Lichtenstein). Assessments should be:

-consistent, they should not vary with the assessment method, nor over time (assuming the assessor gets no new information),
-coherent, they should obey the laws of probability (e.g. Bayes' rule),
-informative, they should contain information about actual outcome values of the quantities assessed,
-well-calibrated, in the long run assessed probabilities should approximate empirical frequencies of outcomes.

In this bachelor thesis we are concerned with the last criterion (calibration). There are different ways to quantify this criterion, for example the calibration score using the chi-squared distribution (Cooke, Mendel, and Thijs, 1988), the calibration score using the binomial distribution (Hanea and Nane, 2019) and the Brier score (Brier, 1950). In the next two subsubsections the calibration score using the chi-squared distribution and the calibration score using the

binomial distribution will be discussed. The Brier score BS is defined as

$$BS = \frac{1}{n}\sum_{t=1}^{n}(f_t - o_t)^2,$$

where f_t is the probability that was given by the expert and o_t the actual outcome of event t (where $o_t = 1$ if the event did occur and $o_t = 0$ if the event did not occur) (Brier, 1950).

2.2.1 Calibration score using chi-squared distribution

In Cooke's method (Cooke, 1991) the difference between s_i and p_i is measured using the relative information of s_i with respect to p_i. That is

$$I(s_i, p_i) = s_i \ln \frac{s_i}{p_i} + (1 - s_i) \ln \frac{1 - s_i}{1 - p_i}$$

for $i \in \{1, 2, ..., 10\}$. In Cooke's method, $\sum_{i=1}^{10} 2n_i I(s_i, p_i) \sim \chi^2_{10}$. The calibration score of expert e is then defined as

$$cal_{\chi^2}(e) = 1 - F(\sum_{i=1}^{10} 2n_i I(s_i, p_i)),$$

where F is the cumulative distribution function of the chi-square distribution with 10 degrees of freedom. The reasoning behind this calibration score can be found in (Cooke, 1991). Every calibration score takes values from 0 to 1 and a small difference between s and p gives a small relative information of s with respect to p and thus a high calibration score. $cal_{\chi^2}(e)$ works best for a sufficient amount of questions, where a rule of thumb for the amount of calibration questions is proposed in (Bhola and Cooke, 1992), which states that the amount of calibration questions need to satisfy

$n_i p_i \geq 4$ and $n_i(1 - p_i) \geq 4$ for each i.

This rule of thumb implies that at least 80 events need to be assigned to bins B_1 and B_{10}, at least 27 events need to be assigned to bins B_2 and B_9, at least 16 events need to be assigned to bins B_3 and B_8, at least 12 events need to be assigned to bins B_4 and B_7 and at least 9 events need to be assigned to bins B_5 and B_6 in order to obtain a reliable calibration score. Besides the unlikely criterion that an expert answers the calibration questions in such a way that the described bin assignment occurs, also there are at least 288 calibration

questions needed. In order to be able to use Cooke's calibration score in practice, in most cases the number of bins used for the expert judgement need to be reduced.

2.2.2 Calibration score using binomial distribution

In order to use ten bins and to avoid Bhola's and Cooke's rule of thumb, Hanea and Nane have developed a calibration score based on the binomial distribution (Hanea and Nane, 2019). This calibration score uses that under the null hypothesis $H_0 : s = p$, $s_i n_i$ follows a binomial distribution with parameters n_i and p_i, that is, $s_i n_i \sim Bin(n_i, p_i)$. Let $Y_i \sim Bin(n_i, p_i)$ and $Y = \sum_{i=1}^{10} Y_i$. Hanea and Nane propose a calibration score based on this distribution, where the two-sided mid-p-value of the hypothesis test,

$$\pi_{two}(a) = 2 \cdot \min(P(Y > a) + 0.5P(Y = a), P(Y < a) + 0.5P(Y = a))$$

is choosen. For more information about the two-sided mid-p-value, see (Lancaster, 1949; Lancaster, 1961; Agresti, 2003). The calibration score of an expert e is then defined as

$$Cal_{bin}(e) = \pi_{two}(\sum_{i=1}^{10} n_i s_i)$$

3 Some properties of calibration scores for discrete random variables using simulated data

Hanea and Nane concluded in their paper (Hanea and Nane, 2019) that after investigating their calibration score's theoretical properties and practical performance they found this score has a number of positive and negative attributes. An example of a positive attribute is that his calibration score relies on an exact distribution, which will hopefully significantly reduce the necessary number of questions needed in order to provide reliable scores. An example of a negative attribute is that that this calibration score sometimes gives inexplicably small values.
In this section, the focus will be on looking at different properties of this calibration score using cases where all events get assigned to the same bin to hopefully find a way to get rid of some of the negative attributes. Also, in this section, other calibration scores which are slight adjustments of Hanea's and Nane's calibration score are introduced as well as looking at the properties of these calibration scores using cases where all events got assigned to the same bin. The code used in this section can be found in appendix B.

3.1 All events assigned to same bin

In this subsection, calibration scores are computed for situations where an expert assigns all events to the same bin. In table 1, the following results can be seen for Cooke's calibration score and for Hanea's and Nane's calibration score (this results were given to me by G. F. Nane, except for the last two rows).
As can be seen in table 1, Hanea's and Nane's calibration score takes lower values than Cooke's calibration score for these simulations. In row 2 and row 3 of table 1, a calibration score close to 1 is desired. This is the case since the expert assigned all events to bin B_{10} and all events indeed occured. Cooke's calibration score in these two cases is very close to 1, but Hanea's and Nane's calibration score takes the values 0.35849 and 0.21464, which is not the desired result.
In the fourth row of table 1, a calibration score close to value 1 is desired, as all events are assigned to the fifth bin and one out of two events occur. Again,

# questions	Bin assignment	s_i	Chi	Bin
20	(0,0,0,0,0,0,0,0,0,20)	(0,0,0,0,0,0,0,0,0,1)	0.99062	0.35849
30	(0,0,0,0,0,0,0,0,0,30)	(0,0,0,0,0,0,0,0,0,1)	0.961138	0.21464
30	(0,0,0,0,30,0,0,0,0,0)	(0,0,0,0,0.5,0,0,0,0,0)	0.999996	0.58606
30	(0,0,30,0,0,0,0,0,0,0)	(0,0,0.3,0,0,0,0,0,0,0)	1	0.86234
30	(0,0,0,0,30,0,0,0,0,0)	(0,0,0,0,0.3,0,0,0,0,0)	0.995448	0.20445
30	(0,0,0,0,30,0,0,0,0,0)	(0,0,0,0,0.2,0,0,0,0,0)	0.5087536	0.00507
30	(0,0,0,0,30,0,0,0,0,0)	(0,0,0,0,0.1,0,0,0,0,0)	0.04051272	5e-05
30	(0,0,0,0,0,0,0,30,0,0)	(0,0,0,0,0,0,0,0.8,0,0)	0.9999856	0.55065
30	(0,0,0,0,0,0,0,30,0,0)	(0,0,0,0,0,0,0,0.6,0,0)	0.9536956	0.07225
30	(0,0,0,0,0,0,0,30,0,0)	(0,0,0,0,0,0,0,0.5,0,0)	0.9536956	0.07225
30	(0,0,0,0,0,0,0,0,0,30)	(0,0,0,0,0,0,0,0,0,0)	0	0
30	(30,0,0,0,0,0,0,0,0,0)	(1,0,0,0,0,0,0,0,0,0)	0	0

Table 1: Cooke's calibration score and Hanea's and Nane's calibration score for given bin assignment.

Cooke's calibration score is close to 1 but Hanea's and Nane's calibration score takes value 0.58606, which is not the desired result. In the second last row, all events are assigned to bin 10 and zero events occur. Both Cooke's calibration score as well as Hanea's and Nane's calibration score correctly take value 0. In the last row, all events are assigned to bin 1 and all events occur. Again, both Cooke's calibration score as well as Hanea's and Nane's calibration score correctly take value 0.

Using similar reasoning as before, in the fifth row of table 1, a calibration score close to 1 is desired, Cooke's calibration score is exactly 1 whereas Hanea's and Nane's calibration score is 0.85234. Note that in this row Hanea's and Nane's calibration score performs better than in previous rows.

In the sixth row, Cooke's calibration score is again close to 1. Since in this case all events are assigned to the fifth bin but only three out of ten events occur, a calibration score close to 1 is a bit too optimistic. Now Hanea's and Nane's calibration score is 0.20445. This clearly indicates that indeed the performance of this expert was not close to perfect, but whether that is about the right score or a bit too negative is not entirely clear to me.

In row 7, again the expert assigns all events to the fifth bin, this time only one out of five events occur. Cooke's calibration score now is 0.508736, clearly indicating that the expert performed not too well. Hanea's and Nane's calibration score this time is 0.00507, In row 8, again all events are assigned to the the fifth bin, only one out ten events occur. The expert performed poorly

and this is clearly indicated by both Cooke's calibration score and Hanea's and Nane's calibration score, being 0.04051272 and 5e-05 respectively.
Looking at row 9, all events are assigned to the eight bin and eight out of ten events occur. A calibration score close to 1 is desired. Cooke's calibration score is 0.9999856 and Hanea's and Nane's calibration score is 0.55065. Again, Hanea's and Nane's calibration score is a bit on the low side.
In the second last row, all events are assigned to the tenth bin but 0 events occur. A calibration score close to 0 is desired. Both Cooke's calibration score and Hanea's and Nane's calibration score is 0, as desired. In the last row, all evnets are assigned to the first bin and all events occur. Again, a calibration score close to 0 is desired and both Cooke's calibration score and Hanea's and Nane's calibration score is 0.
In summary, Cooke's calibration score performs in most rows better than Hanea's and Nane's calibration score for these simulations.

In order to determine what causes Hanea's and Nane's score to sometimes give a bit of a too low calibration score, each step of the computation of Hanea's and Nane's calibration score is inspected. For this, we take the case where all 20 events get assigned to bin 10 and all events occur, that is, n_i = (0,0,0,0,0,0,0,0,0,20) and s_i = (0,0,0,0,0,0,0,0,0,1). Clearly, element wise multiplication gives $n_i s_i$ = (0,0,0,0,0,0,0,0,0,20) and then summing over the elements of $n_i s_i$ gives 20. Looking at the formulas of subsubsection 2.2.2, for this case, a = 20. Now we look at ($P(Y > a)$, $P(Y < a)$ and $P(Y = a)$. Since for any binassignment, $a \leq 20$ and in this case $a = 20$, if follows that ($P(Y > a) = 0$. Since any cumulative distribution function can only take values in the interval [0, 1], $0 = (P(Y > a) \leq P(Y < a)$. Since Hanea's and Nane's calibration score is
$2 \cdot \min(P(Y > a) + 0.5P(Y = a), P(Y < a) + 0.5P(Y = a)$, in this case the calibration score is $2 \cdot (P(Y > a) + 0.5P(Y = a)) = P(Y = a) = 0.35849$. This shows why for the case where all 20 events get assigned to the same bin Hanea's and Nane's calibration score does not take a value close to one as desired, but a value of 0.35849.
Let's look at a second case, this time the case that all 30 events get assigned to bin 5 and 50% of the events occur, that is, n_i = (0,0,0,0,30,0,0,0,0,0) and s_i = (0,0,0,0,0.5,0,0,0,0,0). Then element wise multiplication gives $n_i s_i$ = (0,0,0,0,15,0,0,0,0,0) and then summing over the elements of $n_i s_i$ gives 15. Looking at the formulas of subsubsection 2.2.2, for this case, a = 15. Again,

we look at $(P(Y > a)$, $P(Y < a)$ and $P(Y = a)$. $P(Y = a) = 0.1242479$, $P(Y < a) = 0.1354354$ and $(P(Y > a) = 0.8555356$ and so the calibration score in this case is 0.5860639, while a score close to 1 in this case is desired. Analyzing these two cases give the idea that using another formula for the p-value might give better results.

In table 2, Cooke's calibration score and Hanea's and Nane's calibration score are computed for different bin assignments, $s_i's$ and ten calibration questions. The code for these computations can be found in appendix A.

Bin assignment	s_i	Chi	Bin
(0,0,0,0,0,0,0,0,0,10)	(0,0,0,0,0,0,0,0,0,1)	0.9998065	0.59874
(0,0,0,0,10,0,0,0,0,0)	(0,0,0,0,0.5,0,0,0,0,0)	1	0.75716
(0,0,10,0,0,0,0,0,0,0)	(0,0,0.3,0,0,0,0,0,0,0)	1	0.69853
(0,0,0,0,10,0,0,0,0,0)	(0,0,0,0,0.3,0,0,0,0,0)	0.9998683	0.3656
(0,0,0,0,10,0,0,0,0,0)	(0,0,0,0,0.2,0,0,0,0,0)	0.9866848	0.12282
(0,0,0,0,10,0,0,0,0,0)	(0,0,0,0,0.1,0,0,0,0,0)	0.8271783	0.02579
(0,0,0,0,0,0,0,10,0,0)	(0,0,0,0,0,0,0,0.8,0,0)	1	0.76962
(0,0,0,0,0,0,0,10,0,0)	(0,0,0,0,0,0,0,0.6,0,0)	0.9997529	0.30225
(0,0,0,0,0,0,0,10,0,0)	(0,0,0,0,0,0,0,0.5,0,0)	0.984176	0.09785
(0,0,0,0,0,0,0,0,0,10)	(0,0,0,0,0,0,0,0,0,0)	0	0
(10,0,0,0,0,0,0,0,0,0)	(1,0,0,0,0,0,0,0,0,0)	0	0

Table 2: Cooke's calibration score and Hanea's and Nane's calibration score for given bin assignment using 10 questions.

Again, Hanea's and Nane's calibration score gives lower values than desired, though the scores are higher than in table 1. Surprisingly is how high Cooke's calibration score is for all cases using ten calibration questions, except for the last two rows, where Cooke's calibration score correctly takes 3.761608e-09. Those high calibration scores may be caused by the Chi-squared distribution.

3.2 Hanea's and Nane's calibration score using different formulas for p-value

Hanea and Nane used as formula for the p-value the mid-p-value as described in (Rivals et al., 2006) and as stated in subsubsection 2.2.2. In (Rivals et al., 2006), an overview of three other formulas for two-sided tests with a discrete null distribution is given, namely the two-sided p-value defined as twice the one-sided p-value and two formulas using the minimum-likelihood approach.

The formula for the two-sided p-value is

$$p_{two}^{doubling}(a) = 2 \cdot \min(P(Y > a), P(Y < a)).$$

Clearly, for any a, this formula gives values lower or equal than for the mid-p-value, since the mid-p-value also takes two multiplied by the minimum of $P(Y \geq a)$ and $P(Y \leq a)$, but by both components of the minimum $0.5P(Y = a)$ is added. For many rows of table 2, a higher calibration score is desired. Therefore, Hanea's and Nane's calibration score using $p_{two}^{doubling}(a)$ will not solve these cases, but it might work well in different cases, for example when it is applied to actual data. Therefore, the results of this calibration score, in the table denoted as Cal score bin 2, using $p_{two}^{doubling}(a)$ are given in table 3. The case where all events occur and are assigned to bin 10 is notable, as a score close to one is desired but calibration score bin 2 takes a value of zero. In this case we have n_i = (0,0,0,0,0,0,0,0,0,10) and s_i = (0,0,0,0,0,0,0,0,0,1). Clearly, element wise multiplication gives $n_i s_i$ = (0,0,0,0,0,0,0,0,0,10) and then summing over the elements of $n_i s_i$ gives 10. In this case, for the formula of calibration score bin 2, a = 10. Since when assigning ten events to ten bins a can never be greater than 10 and a = 10, $P(Y > a) = 0$ and this explains why calibration score bin 2 takes a value of 0, while a value close to 1 is desired.

Bin assignment	s_i	Bin	Bin 2
(0,0,0,0,0,0,0,0,0,10)	(0,0,0,0,0,0,0,0,0,1)	0.59874	0
(0,0,0,0,10,0,0,0,0,0)	(0,0,0,0,0.5,0,0,0,0,0)	0.75716	0.52313
(0,0,10,0,0,0,0,0,0,0)	(0,0,0.3,0,0,0,0,0,0,0)	0.69853	0.44825
(0,0,0,0,10,0,0,0,0,0)	(0,0,0,0,0.3,0,0,0,0,0)	0.3656	0.19912
(0,0,0,0,10,0,0,0,0,0)	(0,0,0,0,0.2,0,0,0,0,0)	0.12282	0.04651
(0,0,0,0,10,0,0,0,0,0)	(0,0,0,0,0.1,0,0,0,0,0)	0.02579	0.00507
(0,0,0,0,0,0,0,10,0,0)	(0,0,0,0,0,0,0,0.8,0,0)	0.76962	0.48805
(0,0,0,0,0,0,0,10,0,0)	(0,0,0,0,0,0,0,0.6,0,0)	0.30225	0.15625
(0,0,0,0,0,0,0,10,0,0)	(0,0,0,0,0,0,0,0.5,0,0)	0.09785	0.03946
(0,0,0,0,0,0,0,0,0,10)	(0,0,0,0,0,0,0,0,0,0)	0	0
(10,0,0,0,0,0,0,0,0,0)	(1,0,0,0,0,0,0,0,0,0)	0	0

Table 3: Hanea's and Nane's calibration score and calibration score bin 2 for given binassignment using 10 questions.

The two formulas using the minimum-likelihood approach are

$$p_{two}^{minlik}(a) = \sum_{P(Y=b)\leq P(Y=a)} P(Y=b)$$

and

$$\pi_{two}^{minlik}(a) = \sum_{P(Y=b)<P(Y=a)} P(Y=b) + 0.5 \sum_{P(Y=b)=P(Y=a)} P(Y=b).$$

Clearly, $p_{two}^{minlik}(a) \geq \pi_{two}^{minlik}(a)$ for every a, because if $P(Y=b) = P(Y=a)$, in the sum of $\pi_{two}^{minlik}(a)$ we multiply these elements with 0.5, whereas these elements in $p_{two}^{minlik}(a)$ are multiplied with one.
The results of the calibration score, in the table denoted as Cal score bin 3, using $p_{two}^{minlik}(a)$ are given in table 4. Furthermore, calibration score bin 3 seems to perform best in cases where all events are assigned to the same bin compared to the other other calibration scores based on the binomial distribuion.

Bin assignment	s_i	Bin	Bin 3
(0,0,0,0,0,0,0,0,0,10)	(0,0,0,0,0,0,0,0,0,1)	0.59874	1
(0,0,0,0,10,0,0,0,0,0)	(0,0,0,0,0.5,0,0,0,0,0)	0.75716	0.76163
(0,0,10,0,0,0,0,0,0,0)	(0,0,0.3,0,0,0,0,0,0,0)	0.69853	0.71843
(0,0,0,0,10,0,0,0,0,0)	(0,0,0,0,0.3,0,0,0,0,0)	0.3656	0.5276
(0,0,0,0,10,0,0,0,0,0)	(0,0,0,0,0.2,0,0,0,0,0)	0.12282	0.20155
(0,0,0,0,10,0,0,0,0,0)	(0,0,0,0,0.1,0,0,0,0,0)	0.02579	0.02776
(0,0,0,0,0,0,0,10,0,0)	(0,0,0,0,0,0,0,0.8,0,0)	0.76962	1
(0,0,0,0,0,0,0,10,0,0)	(0,0,0,0,0,0,0,0.6,0,0)	0.30225	0.28044
(0,0,0,0,0,0,0,10,0,0)	(0,0,0,0,0,0,0,0.5,0,0)	0.09785	0.13444
(0,0,0,0,0,0,0,0,0,10)	(0,0,0,0,0,0,0,0,0,0)	0	0
(10,0,0,0,0,0,0,0,0,0)	(1,0,0,0,0,0,0,0,0,0)	0	0

Table 4: Hanea's and Nane's calibration score and calibration score bin 3 for given binassignment using 10 questions.

Looking at table 4, in each row Cal score bin 3 is higher than Cal score bin, except for the second last row. Also, for the second and the eight row, since Cal score bin 3 is exactly one, it performs well in these cases. In some cases, there is only a small difference between the two scores, see row 3, 4, 7 and 9.

The results of the calibration score, in the table denoted as Cal score bin 4,

using $\pi_{two}^{minlik}(a)$ are given in table 5. As was stated earlier in this subsection,

Bin assignment	s_i	Cal score bin	Cal score bin 4
(0,0,0,0,0,0,0,0,0,10)	(0,0,0,0,0,0,0,0,0,1)	0.59874	0.70063
(0,0,0,0,10,0,0,0,0,0)	(0,0,0,0,0.5,0,0,0,0,0)	0.75716	0.64462
(0,0,10,0,0,0,0,0,0,0)	(0,0,0.3,0,0,0,0,0,0,0)	0.69853	0.59329
(0,0,0,0,10,0,0,0,0,0)	(0,0,0,0,0.3,0,0,0,0,0)	0.3656	0.44436
(0,0,0,0,10,0,0,0,0,0)	(0,0,0,0,0.2,0,0,0,0,0)	0.12282	0.1634
(0,0,0,0,10,0,0,0,0,0)	(0,0,0,0,0.1,0,0,0,0,0)	0.02579	0.0174
(0,0,0,0,0,0,0,10,0,0)	(0,0,0,0,0,0,0,0.8,0,0)	0.76962	0.85922
(0,0,0,0,0,0,0,10,0,0)	(0,0,0,0,0,0,0,0.6,0,0)	0.30225	0.20744
(0,0,0,0,0,0,0,10,0,0)	(0,0,0,0,0,0,0,0.5,0,0)	0.09785	0.10524
(0,0,0,0,0,0,0,0,0,10)	(0,0,0,0,0,0,0,0,0,0)	0	0
(10,0,0,0,0,0,0,0,0,0)	(1,0,0,0,0,0,0,0,0,0)	0	0

Table 5: Hanea's and Nane's calibration score and calibration score bin 4 for given binassignment using 10 questions.

Cal score bin 4 will always have a smaller value than Cal score bin 3 and this can be seen in tables 4 and 5. Furthermore Cal score bin 4 has four cases where it has a lower value than Cal score bin. Also, Cal score bin 4 performs clearly worse than Cal score bin 3 in the second and eight row, since Cal score bin 3 takes the values 1, whereas Cal score bin 4 takes the values 0.70063 and 0.85922, respectively. However, there can be other cases where Cal score bin 4 might perform better than Cal score bin 3.

3.3 Calibration score and sample size

In order to check if the different calibration scores are affected by the sample size, each calibration score is computed for similar situations using 10, 30, 50 and 100 calibration questions. The values of the calibration scores can be found in the tables of subsection 3.1, 3.2 and appendix A.
There are two cases where for all four sample sizes, all five calibration scores are zero. These two cases are when all events are assigned to bin 10 but zero events occur and when all events are assigned to bin 1 and all events occur. Note that a score of 0 for these two cases is good. For calibration score bin 2, there is another case where for each sample size the score is zero. This is the case where all events get assigned to bin 10 and all events occur. Note that a score close to 1 is desired. The reason why calibration score bin 2 takes a value of 0 in this case is treated in subsection 3.2.

For Cooke's calibration score, in 6 out of 11 cases the difference between scores for different sample sizes is greater than 0.1. For Hanea's and Nane's calibration score, this difference occurs in 7 out of 11 cases, for calibration score bin 2, this difference occurs in 5 out of 11 cases for calibration score bin 3 and calibration score bin 4.
The number of cases where a calibration score is strictly decreasing as the sample size increases happens for Cooke's calibration score in 9 out of 11 cases, for Hanea's and Nane's calibration score in 8 out of 11, for calibration score bin 2 in 7 out of 11, for calibration score bin 3 in 9 out of 11 cases and for calibration score bin 4 in 9 out of 11 cases. In some cases, this (sometimes) strong decrease might cause each calibration score to give relative low scores compared to a value close to one which is in some cases desired.
For cases where all events got assigned to the same bin, calibration score bin 3 seems to perform best when 10, 30 and 50 calibration questions were used, as it most frequently took the closest value of one when a score close to 1 was desired compared to the other calibration scores based on the binomial distribution and correctly displays it when an expert did not answer the calibration questions perfectly. For 100 calibration questions, the difference between Hanea's and Nane's calibration score is always smaller than 0.1 and so they give close values compared to each other.

3.4 Summarizing results

Summarizing the important points and results, calibration score bin 2 takes always lower or equal values than Hanea's and Nane's calibration score and calibration score bin 4 takes always lower or equal values than calibration score bin 3.
Also, it is inspected if the different calibration scores are affected by the sample size, where each calibration score is computed for similar situations using 10, 30, 50 and 100 calibration questions. It turns out that as the sample size increases, each calibation score tends to strictly decrease. In cases where a score close to 1 is desired, sometimes using a larger sample size makes each calibration score perform worse. Also, in most cases which were treated in this section, for each calibration score, the difference between scores using different sample sizes is greater than 0.1.
For cases where all events got assigned to the same bin, calibration score bin 3 seems to perform best when 10, 30 and 50 calibration questions were used. For

100 calibration questions Hanea's and Nane's calibration score and calibration score bin 3 for all cases took values close to each other.

4 Some properties of calibration scores for discrete random variables with data

In practice, it does not often occur that all events get assigned to the same bin. Therefore, in this section, data is used where the data consists of experts answering calibration questions. Three different data files are used in this bachelorthesis referred to as dataSIPS, dataACE_IDEA and dataACE_GJP. In the next three subsections, information about the data files and the behaviour of the different calibration scores for each datafile can be found. The code used for this subsection can be found in appendix C.

4.1 DataSIPS

The data is from the repliCATS project from the university of Melbourne. The aim of this project is to use crowdsource to evaluate the credibility of published research in business research, criminology, economics, education, political science, psychology, public administration, and sociology (the repliCATS project, z.d.-b). The datafile DataSips consists of 25 experts each answering 25 calibration questions. Note that since each expert answered 25 calibration questions and ten bins are used, this does not satisfy a sufficient amount of calibration questions needed for Cooke's calibration to be robust according to a rule of thumb as proposed in (Bhola and Cooke, 1992), which should at least be 288 calibration questions (even if the other criterion of the rule of thumb which gives conditions to how many events at least should be assigned to each bin is ignored). Therefore, this data does not contain cases where the calibration scores based on a binomial distribution can be compared to Cooke's calibration score which is guaranteed to be robust.

4.1.1 Behaviour of calibration scores

In figure 2, the plot of the five different calibration scores for each expert can be found. It can be seen that calibration score bin 2 always takes lower values than the other three calibration scores based on a binomial distribution. Only for two experts, Cooke's calibration score gives lower values than calibration score bin 2. Calibration score bin 3 gives except for one expert the highest score compared to the other three calibration scores based on the binomial distribution. There is one expert where Hanea's and Nane's calibration score

gives a higher score than calibration score bin 3. Cooke's calibration score gives most frequently the highest score, that is, for 13 out of 25 experts. The difference between Hanea's and Nane's calibration score and calibration score bin 4 is for seven out of twentyfive experts greater than 0.05 and the difference is never greater than 0.1. Therefore, calibration score bin 4 often gives results close to Hanea's and Nane's calibration score. For 22 out of 25 experts, the difference between Cooke's calibration score and calibration score bin 3 is greater than 0.05. For 20 out of 25 experts, this difference is greater than 0.1. Therefore, calibration score bin 3 does rarely give results close to Cooke's calibration score.

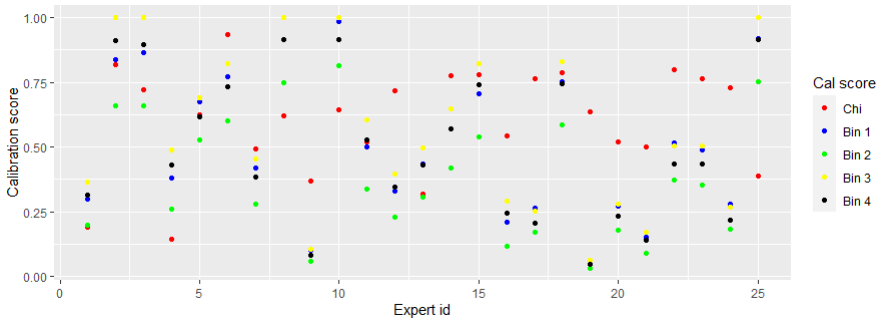

Figure 2: Plot of each calibration score for each expert in dataSIPS

In tables 6, for each expert used in the dataSIPS file, the expert id and the five different calibration scores can be found. It can be seen that for the expert with id 19, Cooke's calibration score is 0.6346904. Calibration score bin 1, bin 2, bin 3 and bin 4 take values 0.06262, 0.03329, 0.06157 and 0.04691. Since for this expert, $(n_1, n_2, n_3, n_4, n_5, n_6, n_7, n_8, n_9, n_{10}) = (2\ 1\ 3\ 2\ 1\ 1\ 3\ 5\ 4\ 3)$ and $(s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9, s_{10}) = (0, 0, 0, 0.5, 1, 1, 0.6666667, 0.6, 0.75, 0.6666667)$, this expert did not perform so bad that a calibration score below 0.1 seems justifiable and so Cooke's calibration score seems more appropriate. Calibration score bin 3 takes values 1 for experts with id 10, 8, 25, 3 and 2. This means that calibration score bin 3 states that these experts answered the calibration questions perfectly. In order to see if this is true or whether this score sometimes states that the experts answered the calibration questions perfectly while this is not the case, again we inspect the different n_i's and s_i's for the different experts, which can be found in table 7.

Expert id	Chi	bin	bin 2	bin 3	bin 4
12	0.7195363	0.32975	0.22829	0.39732	0.34659
22	0.8010957	0.51625	0.37363	0.50531	0.434
10	0.6441146	0.98616	0.81629	1.00000	0.91506
6	0.9332712	0.77394	0.60069	0.82194	0.73531
5	0.6246355	0.67463	0.52794	0.68934	0.61599
18	0.7866629	0.75111	0.58687	0.82861	0.74649
15	0.7782713	0.70478	0.53997	0.82194	0.73954
16	0.5421891	0.20988	0.11743	0.29062	0.24439
11	0.5205494	0.49936	0.33902	0.60657	0.52639
21	0.4998845	0.15089	0.09064	0.17043	0.14030
4	0.1432706	0.38143	0.25968	0.49033	0.42946
14	0.7765745	0.57083	0.41814	0.64649	0.57015
8	0.6192868	0.91452	0.74732	1.00000	0.91640
20	0.5199728	0.27228	0.17944	0.27777	0.23135
25	0.3883761	0.91900	0.75394	1.00000	0.91747
19	0.6346904	0.06262	0.03329	0.06157	0.04691
3	0.7199183	0.86337	0.65974	1.00000	0.89818
2	0.8173941	0.83746	0.66135	1.00000	0.91194
7	0.4913925	0.41895	0.28010	0.45532	0.38589
9	0.3702145	0.10220	0.05765	0.10558	0.08331
23	0.7637622	0.49024	0.35305	0.50374	0.43515
17	0.7658761	0.26540	0.17061	0.25389	0.20650
24	0.7312899	0.27779	0.18198	0.26700	0.21909
1	0.1886612	0.30034	0.19959	0.36629	0.31591
13	0.3165419	0.43601	0.30608	0.49715	0.43219

Table 6: Calibration scores Chi, bin and bin 2 for SIPS data.

Expert id	n_i	s_i
10	(0, 1, 0, 1, 2, 7, 4, 2, 7, 1)	(0, 0, 0, 0, 0, 0.5714286, 0.25, 1, 0.7142857, 1)
8	(6, 5, 0, 1, 0, 4, 3, 1, 5, 0)	(0.1666667, 0.4, 0, 0, 0, 0.75, 0.3333333, 1, 1, 0)
25	(0, 0, 2, 4, 7, 5, 3, 4, 0, 0)	(0, 0, 0, 0, 0.4285714, 0.6, 1, 1, 0, 0)
3	(0, 2, 2, 2, 3, 7, 6, 2, 1, 0)	(0, 0, 0, 0, 0, 0.7142857, 0.5, 1, 1, 0)
2	(0, 3, 1, 1, 4, 1, 5, 7, 3, 0)	(0, 0, 0, 0, 0.25, 1, 0.6, 0.7142857, 1, 0).

Table 7: Bin assign and empirical probability vector for different experts.

For experts with id 8, 10 and 25, this calibration score of 1 is overly positive. The experts with id 2 and 3 did not perform perfect, but very good, and so a calibration score of 1 is justifiable.

4.2 For dataACE_IDEA

The data is from the research article (Hanea et al., 2021). The research in this article is based upon work funded by DARPA, which stands for Defense Advanced Research Project Agency. DARPA is an agency of the United States Department of Defense responsible for the development of technologies for use by the military. The datafile dataACE_IDEA consists of 150 experts, where experts answered a different amount of questions, ranging from just one up to 96 calibration questions. Note that since the experts answered at most 96 calibration questions, this does not satisfy a sufficient amount of calibration questions needed for Cooke's calibration score to be robust according to a rule of thumb as proposed in (Bhola and Cooke, 1992), which should at least be 288 calibration questions (even if the other criterion of the rule of thumb which gives conditions to how many events at least should be assigned to each bin is ignored).

4.2.1 Behaviour of calibration scores

Using calibration score bin 2, 40 out of 150 experts got a score of zero. In table 8, all other calibration scores never take values below 0.5, but calibration score bin 2 takes in every row value zero (not displayed in the table). This might seem as that calibration score bin 2 is too low. However, since the number of questions ranges from one up to and including four which is a very low number of questions this is not enough to conclude that calibration score bin 2 gives too low values. Note that in all these rows Cooke's calibration score take values close to 1. Also, using calibration score bin 3, 1 out of 150 experts got a score of 0.
Using calibration score bin 3, 34 out of 150 experts got a score of one. In table 9, cal score bin 3 in all four cases takes value one, but all other calibration scores take vakues lower than 0.9. This might seem as that calibration score 3 is too high, but since this are 4 out of 150 cases this is not enough to conclude that calibration score bin 3 gives too high values.
For 96 out of 150 experts, the difference between Cooke's calibration score and calibration score bin 3 is greater than 0.05. For 82 out of 150 experts, this difference is greater than 0.1. Therefore, in most cases, calibration score 3 does not give results close to Cooke's calibration score. Also, using Hanea's and Nane's calibration score, 0 out of 150 experts got a score of one.
For 59 out of 150 experts, the difference between calibration score bin 4 and

Expert id	# questions	Chi	bin	bin 3	bin 4
308	3	9.999963e-01	0.85738	1	0.57131
310	4	9.967547e-01	0.46059	1	0.76970
33	4	9.962073e-01	0.44614	1	0.77693
34	4	9.962073e-01	0.44614	1	0.77693
223	1	9.999445e-01	0.75	1	0.625
187	2	9.990043e-01	0.56250	1	0.71875
241	4	9.999870e-01	0.81451	1	0.59275
57	1	9.996958e-01	0.65	1	0.675
168	3	9.970595e-01	0.46962	1	0.76519
86	1	9.999953e-01	0.85	1	0.575

Table 8: Calibration scores Chi, bin and bin 2 for DATA_ACE data.

Expert id	# questions	Chi	bin	bin 2	bin 4
24	11	0.2526121	0.85940	0.56766	0.85413
85	31	0.3876623	0.78836	0.57791	0.89478
104	23	0.4573048	0.79213	0.59177	0.89982
218	22	0.7841410	0.87597	0.60632	0.86518

Table 9: Calibration scores Chi, bin and bin 2 for DATA_ACE data.

Hanea's and Nane's calibration score is greater than 0.05. For 38 out of 150 experts, this difference is greater than 0.1. Therefore, in most cases calibration score bin 4 gives results close to Hanea's and Nane's calibration score.
For 33 out of 150 experts, the absolute difference between Cooke's calibration score and each calibration score based on the binomial distribution is greater than 0.5.
To get a better idea of the behaviour of the different calibration scores, in figure 3, the plot of the five different calibration scores for each expert who answered at least ten questions can be found. The amount of experts who answered more than nine questions is 84. Again, it can be seen that calibration score bin 4 takes most frequently a value of 0 or a value close to 0. In some of the cases where calibration score bin 4 takes a value close to 0 or is exactly 0, some or all other calibration scores also take values close to 0. Also, calibration score 2 also tends to give relatively low values compared to Cooke's calibration score, Hanea's and Nane's calibration score and calibration score bin 3. Furthermore, calibration score bin 3 takes in almost all cases higher values than the other three calibration scores based on the binomial distribution. Also calibration score bin 3 sometimes takes higher values than

Cooke's calibration score.

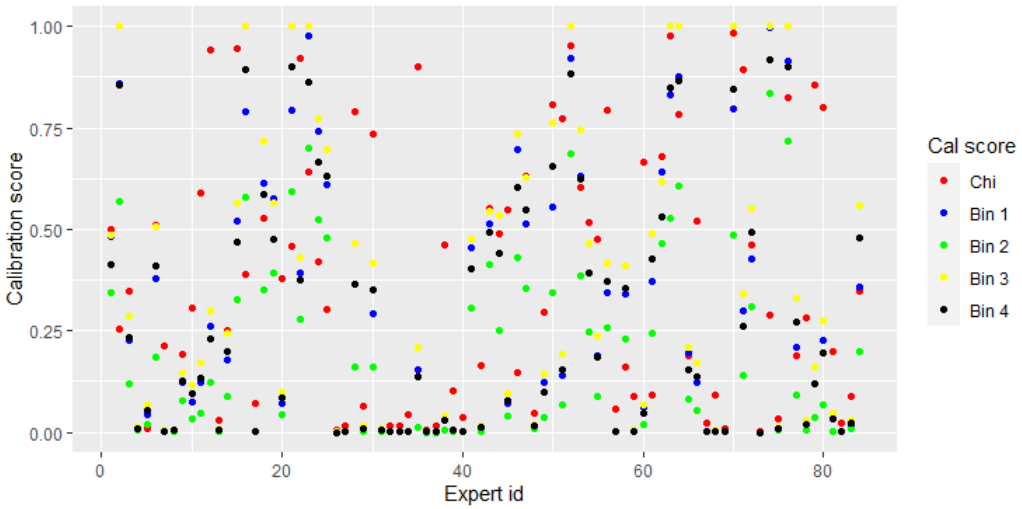

Figure 3: Plot of each calibration score for each expert in dataACE_IDEA answering 10 up to and including 100 questions

4.3 For dataACE_GJP

The data is from The Good Judgement Project. This organization uses forecasting for example to predict topics with regards to the USA election or topics with regards to the Russia - Ukraine war (Good Judgment Inc, 2023). The datafile dataACE_GJP consists of 4844 experts, where experts answered a different amount of questions, ranging from just one up to 256 calibration questions. Note that since the experts answered at most 256 calibration questions, this does not satisfy a sufficient amount of calibration questions needed for Cooke's calibration score to be robust according to a rule of thumb as proposed in (Bhola and Cooke, 1992), which should at least be 288 calibration questions (even if the other criterion of the rule of thumb which gives conditions to how many events at least should be assigned to each bin is ignored).

4.3.1 Behaviour of calibration scores

Using calibration score bin 2, 1115 out of 4844 experts got a score of zero. For calibration score bin 4 this is for 273 out of 4844 experts. Therefore, it seems

that calibration score bin 2 tends to give low values. Using calibration score bin 3, 1059 out of 4844 experts got a score of one. For Hanea's and Nane's calibration score, 20 out of 4844 experts got a score of one.
For 2861 out of 4844 experts, the difference between Cooke's calibration score and calibration score bin 3 is greater than 0.05. For 2465 out of 4844 experts, this difference is greater than 0.1. Therefore, in most cases, calibration score 3 does not give results close to Cooke's calibration score.
For 1549 out of 4844 experts, the difference between calibration score bin 4 and Hanea's and Nane's calibration score is greater than 0.05. For 955 out of 4844, this difference is greater than 0.1. Therefore, in most cases calibration score bin 4 often gives results close to Hanea's and Nane's calibration score.
For 6 out of 4844 experts, the absolute difference between Cooke's calibration score and each calibration score based on the binomial distribution is greater than 0.5.
In figure 4, the plot of the five different calibration scores for each expert who answered at least ten and at most 100 questions can be seen. Since the amount of experts who answered at least ten and at most 100 questions is 2778, the plot is a bit cluttered. However, still the behaviour of the different calibration scores can be observed. It can be seen that calibration score bin 4 most frequently values of zero or close to zero. Most frequently, calibration score bin 3 takes the highest value compared to the other calibration scores based on the binomial distribution. Calibration score bin 2 tends to give relatively low values, sometimes taking lower values than calibration score bin 4.

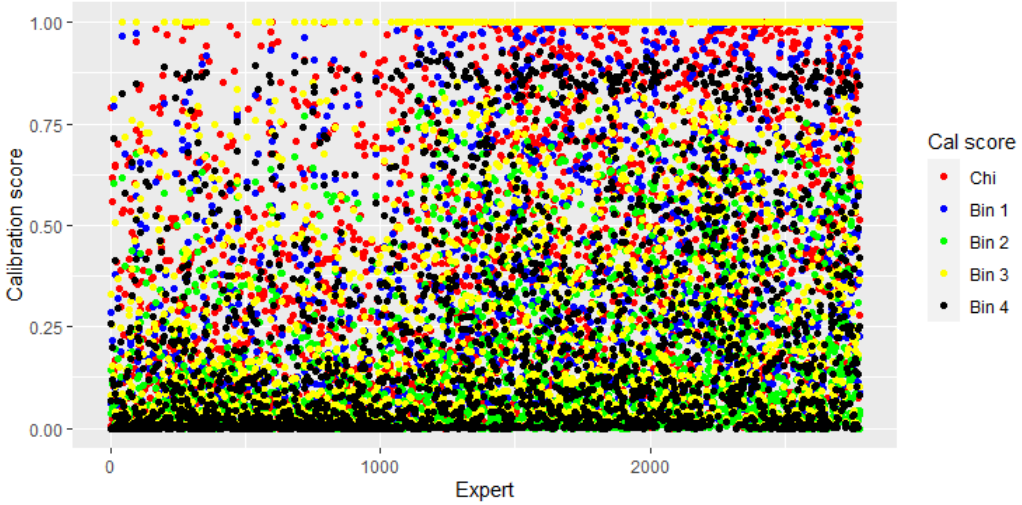

Figure 4: Plot of each calibration score for each expert in dataACE_GJP answering 10 up to and including 100 questions

In figure 5, the plot of the five different calibration scores for each expert who answered more than 100 questions can be seen. The amount of experts who answered more than 100 questions is 518. It can be seen that calibration score bin 4 often takes values of zero or close to zero. This time, there are only four cases where calibration score bin 3 takes values of one or close to one. This is interesting, since for this plot only experts who answered over 100 questions were used and so uncertainty plays a smaller role compared to when experts answered less questions. Bin score 3 still most frequently gives higher values than the other three calibration scores based on the binomial distibution. Calibration score bin 2 tends to give lower values compare to Cooke's calibration score, Hanea's and Nane's calibration score and calibration score bin 3, but in some cases not as low or lower than calibration score bin 4.

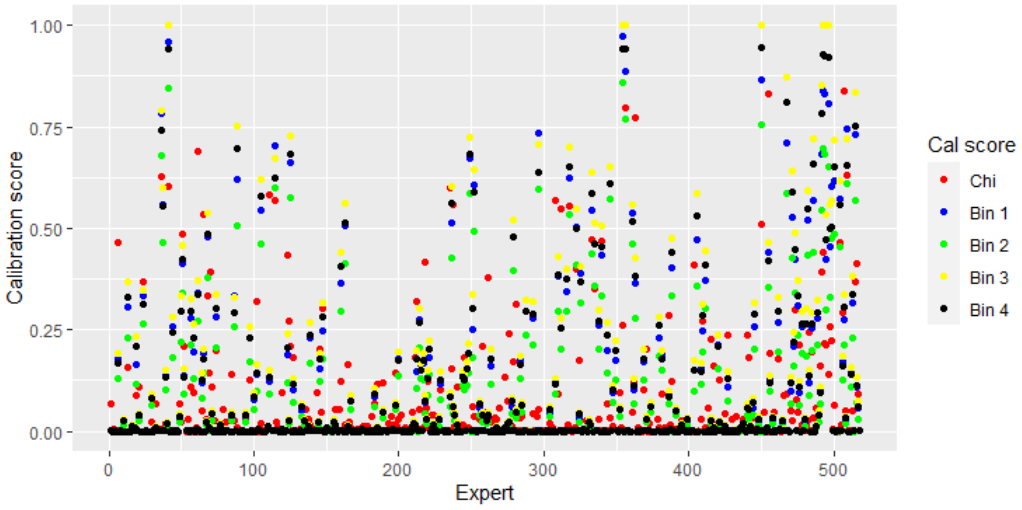

Figure 5: Plot of each calibration score for each expert in dataACE_IDEA answering more than 100 questions

4.4 Summarizing results

In summary, calibration score bin 4 most frequently takes values of zero or close to zero. Also, calibration score bin 2 and 4 tend to give values lower than Cooke's calibration score, Hanea's and Nane's calibration score and calibration score bin 3. Calibration score bin 3 tends to give the highest values compared to the other three calibration scores, frequently taking values of one or close to one, except for dataACE_GJP for all experts answering more than 100 questions, where it happened in 4 out of 518 cases that calibration score bin 3 takes a value of one or close to one. For 33 out of 150 experts from dataACE_IDEA, the absolute difference between Cooke's calibration score and each calibration score based on the binomial distribution is greater than 0.5. However, since 66 out of 150 experts answered less than ten questions, uncertainty might play a big role. For 6 out of 4844 experts from dataACE_GJP, the absolute difference between Cooke's calibration score and each calibration score based on the binomial distribution is greater than 0.5. In this datafile, 2778 out of 4844 experts answered between 10 and 100 questions and 518 out of 4844 experts answered over 100 questions. This means that 1548 out of 4844 experts answered less than 10 questions. Why the

amount of experts whose absolute difference between Cooke's calibration score and each calibration score based on the binomial distrubtion between dataACE_IDEA and dataACE_GJP is so different is not clear to me.

5 Concluding remarks

The behaviour of Hanea's and Nane's calibration score, calibration score bin 2 (which uses the two-sided p-value), calibration score bin 3 (which uses the minimum-likelihood approach) and calibration score bin 4 (which uses another minimum-likelihood approach) is investigated using cases where all events got assigned to the same bin as well as real data where experts gave an estimation for the probability that certain events would occur.
For cases where all events got assigned to the same bin, calibration score bin 3 seems to perform best compared to the other calibration scores based on the binomial distribution. In order to make this statement more robust, more cases where all events got assigned to the same bin must be investigated. Also there are cases where it is hard to distinguish if a score of 0.4 or a score of 0.6 is more suitable for how well the expert answered the calibration questions.
Looking at the behaviour of the different calibration scores based on the binomial distribution using real data, calibration score bin 3 tends to give the highest values compared to the other three calibration scores based on the binomial distribution, most frequently taking values close to or equal to one. However, for experts who answered over 100 calibration questions, calibration score 3 took in only 4 out of 518 cases a calibration score close to or exactly equal to one. Calibration score bin 2 most frequently takes values close to or equal to zero. Also, calibration score bin 2 and 4 tend to give lower values than Cooke's calibration score, Hanea's and Nane's calibration score and calibration score bin 3. This does not come as a total surprise, as calibration score bin 2 by definition always takes values lower or equal than Hanea's and Nane's calibration score and calibration score bin 4 by definition always takes values lower or equal than calibration score bin 3. Also, the absolute difference between Cooke's calibration score and each calibration score based on a binomial distribution being greater than 0.5 is for the dataSIPS file 2 out of 25 experts, for the dataACE_IDEA file 33 out of 150 experts and for the dataACE_GJP file 6 out of 4844 experts. What causes this great absolute difference between each calibration score based on the binomial distribution and Cooke's calibration score for 33 out of 150 experts from dataACE_IDEA is not clear for me.
Since no expert in any three of the datafiles answered 288 or more calibration questions, there is no case that satisfies a sufficient amount of calibration questions needed for Cooke's calibration score to be robust acoording to a rule

of thumb as proposed as proposed in (Bhola and Cooke, 1992) (even if the other criterion of the rule of thumb which gives conditions to how many events at least should be assigned to each bin is ingored). This makes giving statements about which calibration score based on the binomial distribution performs best for the given data difficult.
However, we can still look at the behaviour of the different calibration scores and make statements about which seems to perform best. As calibration score bin 2 by definition gives a score of 0 when all events get assigned to bin 10 (in this case a score close to 1 is desired) I would not say that calibration score 2 seems to perform best. Also, Hanea and Nane (2019) found that their calibration score sometimes gives inexplicably small values and therefore I would not recommend this calibration score in its current form. Calibration scores bin 2 and 4 tend to give relatively low values compared to the other scores. Also, calibration score 3 seems to perform best for the simulated data used during this project. Therefore, although calibration score 3 might sometimes gives a bit too high values in situations where given the bin assignment a lower score might be more suitable, I would say that calibration score bin 3 seems to perform best compared to the other calibration scores based on the binomial distribution.

6 Future research

As feature research, the behaviour of the calibration scores using different number of bins (for example five or 20 bins) could be looked at. Also, determining what is the best way to validate the quality of the assessments of the different experts could be investigated.

References

Agresti, Alan, 2003. Categorical Data Analysis, vol. 482 John Wiley & Sons.

Bhola, B., Cooke, R.M., (1992). Expert opinion in project management. Eur. J. Oper. Res. 57, 24–31.

Brier, G.W., (1950). Verification of forecasts expressed in terms of probability. Mon. Weather Rev. 78, 1–3.

Cooke, R. M. (1991). Experts in uncertainty: opinion and subjective probability in science. Oxford University Press on Demand.

Cooke, R. M., Mendel, M., & Thijs, W. (1988). Calibration and information in expert resolution; a classical approach. Automatica, 24(1), 87-93.

Defense Advanced Research Projects Agency. (z.d.). https://www.darpa.mil/

Good Judgment Inc. (2023, 31 mei). See the future sooner with Superforecasting — Good Judgment. Good Judgment. https://goodjudgment.com/

Hanea, A. M., & Nane, G. F. (2019). Calibrating experts' probabilistic assessments for improved probabilistic predictions. Safety science, 118, 763-771.

Hanea, A. M., Wilkinson, D. P., McBride, M. W., Lyon, A. L., Van Ravenzwaaij, D., Thorn, F. S., Gray, C., Mandel, D. R., Willcox, A., Gould, E., Smith, E. J., Mody, F., Bush, M. B., Fidler, F., Fraser, H., & Wintle, B. C. (2021). Mathematically aggregating experts' predictions of possible futures. PLOS ONE, 16(9), e0256919. https://doi.org/10.1371/journal.pone.0256919
Lancaster, H.O., 1949. The combination of probabilities arising from data in discrete distributions. Biometrika 36 (3/4), 370–382.

Lancaster, H.O., 1961. Significance tests in discrete distributions. J. Am. Stat. Assoc. 56 (294), 223–234.

Morgan, M. G., Henrion, M., & Morris, S. C. (1979). Expert Judgments for Policy Analysis: Report of an Invitational Workshop Held at Brookhaven National Laboratory, 1979, July 8-11, to Explore Problems and Research Needs in Eliciting and Using Subjective Probabilistic Expert Judgments for Policy Analysis Involving Energy and Environmental Systems.

Rivals, I., Personnaz, L., Taing, L., Potier, M.-C., 2006. Enrichment or depletion of a go category within a class of genes: which test? Bioinformatics 23 (4), 401–407.

the repliCATS project. (z.d.-b). https://replicats.research.unimelb.edu.au/

A More tables used in section 2

Bin assignment	s_i	Chi	Bin
(0,0,0,0,0,0,0,0,0,50)	(0,0,0,0,0,0,0,0,0,1)	0.8823728	0.07694
(0,0,0,0,50,0,0,0,0,0)	(0,0,0,0,0.5,0,0,0,0,0)	0.9999932	0.48059
(0,0,50,0,0,0,0,0,0,0)	(0,0,0.3,0,0,0,0,0,0,0)	0.9999785	0.415
(0,0,0,0,50,0,0,0,0,0)	(0,0,0,0,0.3,0,0,0,0,0)	0.9092397	0.03233
(0,0,0,0,50,0,0,0,0,0)	(0,0,0,0,0.2,0,0,0,0,0)	0.1843731	0.00026
(0,0,0,0,50,0,0,0,0,0)	(0,0,0,0,0.1,0,0,0,0,0)	0.00112149	0
(0,0,0,0,0,0,0,50,0,0)	(0,0,0,0,0,0,0,0.8,0,0)	0.9999672	0.42589
(0,0,0,0,0,0,0,50,0,0)	(0,0,0,0,0,0,0,0.6,0,0)	0.8620486	0.02018
(0,0,0,0,0,0,0,50,0,0)	(0,0,0,0,0,0,0,0.5,0,0)	0.1561812	0.00016
(0,0,0,0,0,0,0,0,0,50)	(0,0,0,0,0,0,0,0,0,0)	0	0
(50,0,0,0,0,0,0,0,0,0)	(1,0,0,0,0,0,0,0,0,0)	0	0

Table 10: Cooke's calibration score and Hanea's and Nane's calibration score for given bin assignment using 50 questions.

Bin assignment	s_i	Chi	Bin
(0,0,0,0,0,0,0,0,0,100)	(0,0,0,0,0,0,0,0,0,1)	0.418101	0.00592
(0,0,0,0,100,0,0,0,0,0)	(0,0,0,0,0.5,0,0,0,0,0)	0.9998239	0.3173
(0,0,100,0,0,0,0,0,0,0)	(0,0,0.3,0,0,0,0,0,0,0)	0.9994723	0.25333
(0,0,0,0,100,0,0,0,0,0)	(0,0,0,0,0.3,0,0,0,0,0)	0.491397	0.0023
(0,0,0,0,100,0,0,0,0,0)	(0,0,0,0,0.2,0,0,0,0,0)	0.002158577	0
(0,0,0,0,100,0,0,0,0,0)	(0,0,0,0,0.1,0,0,0,0,0)	6.767618e-09	0
(0,0,0,0,0,0,0,100,0,0)	(0,0,0,0,0,0,0,0.8,0,0)	0.9992134	0.24836
(0,0,0,0,0,0,0,100,0,0)	(0,0,0,0,0,0,0,0.6,0,0)	0.3714682	0.00101
(0,0,0,0,0,0,0,100,0,0)	(0,0,0,0,0,0,0,0.5,0,0)	0.001358348	0
(0,0,0,0,0,0,0,0,0,100)	(0,0,0,0,0,0,0,0,0,0)	0	0
(100,0,0,0,0,0,0,0,0,0)	(1,0,0,0,0,0,0,0,0,0)	0	0

Table 11: Cooke's calibration score and Hanea's and Nane's calibration score for given bin assignment using 100 questions.

Bin assignment	s_i	Bin	Bin 2
(0,0,0,0,0,0,0,0,0,30)	(0,0,0,0,0,0,0,0,0,1)	0.21464	0
(0,0,0,0,30,0,0,0,0,0)	(0,0,0,0,0.5,0,0,0,0,0)	0.58606	0.46182
(0,0,10,0,0,0,0,0,0,0)	(0,0,0.3,0,0,0,0,0,0,0)	0.86234	0.39319
(0,0,0,0,30,0,0,0,0,0)	(0,0,0,0,0.3,0,0,0,0,0)	0.20445	0.06241
(0,0,0,0,30,0,0,0,0,0)	(0,0,0,0,0.2,0,0,0,0,0)	0.00507	0.00218
(0,0,0,0,30,0,0,0,0,0)	(0,0,0,0,0.1,0,0,0,0,0)	5e-05	1e-05
(0,0,0,0,0,0,0,30,0,0)	(0,0,0,0,0,0,0,0.8,0,0)	0.55065	0.4052
(0,0,0,0,0,0,0,30,0,0)	(0,0,0,0,0,0,0,0.6,0,0)	0.07225	0.04319
(0,0,0,0,0,0,0,30,0,0)	(0,0,0,0,0,0,0,0.5,0,0)	0.07225	0.00164
(0,0,0,0,0,0,0,0,0,30)	(0,0,0,0,0,0,0,0,0,0)	0	0
(10,0,0,0,0,0,0,0,0,0)	(1,0,0,0,0,0,0,0,0,0)	0	0

Table 12: Hanea's and Nane's calibration score and calibration score bin 2 for given binassignment using 30 questions.

Bin assignment	s_i	Bin	Bin 3
(0,0,0,0,0,0,0,0,0,30)	(0,0,0,0,0,0,0,0,0,1)	0.21464	0.40246
(0,0,0,0,30,0,0,0,0,0)	(0,0,0,0,0.5,0,0,0,0,0)	0.58606	0.58784
(0,0,10,0,0,0,0,0,0,0)	(0,0,0.3,0,0,0,0,0,0,0)	0.86234	0.529
(0,0,0,0,30,0,0,0,0,0)	(0,0,0,0,0.3,0,0,0,0,0)	0.20445	0.1408
(0,0,0,0,30,0,0,0,0,0)	(0,0,0,0,0.2,0,0,0,0,0)	0.00507	0.00556
(0,0,0,0,30,0,0,0,0,0)	(0,0,0,0,0.1,0,0,0,0,0)	5e-05	6e-05
(0,0,0,0,0,0,0,30,0,0)	(0,0,0,0,0,0,0,0.8,0,0)	0.55065	0.67446
(0,0,0,0,0,0,0,30,0,0)	(0,0,0,0,0,0,0,0.6,0,0)	0.07225	0.08811
(0,0,0,0,0,0,0,30,0,0)	(0,0,0,0,0,0,0,0.5,0,0)	0.07225	0.00471
(0,0,0,0,0,0,0,0,0,30)	(0,0,0,0,0,0,0,0,0,0)	0	0
(10,0,0,0,0,0,0,0,0,0)	(1,0,0,0,0,0,0,0,0,0)	0	0

Table 13: Hanea's and Nane's calibration score and calibration score bin 3 for given binassignment using 30 questions.

Bin assignment	s_i	Bin	Bin 4
(0,0,0,0,0,0,0,0,0,30)	(0,0,0,0,0,0,0,0,0,1)	0.21464	0.29514
(0,0,0,0,30,0,0,0,0,0)	(0,0,0,0,0.5,0,0,0,0,0)	0.58606	0.52572
(0,0,10,0,0,0,0,0,0,0)	(0,0,0.3,0,0,0,0,0,0,0)	0.86234	0.4641
(0,0,0,0,30,0,0,0,0,0)	(0,0,0,0,0.3,0,0,0,0,0)	0.20445	0.1217
(0,0,0,0,30,0,0,0,0,0)	(0,0,0,0,0.2,0,0,0,0,0)	0.00507	0.00411
(0,0,0,0,30,0,0,0,0,0)	(0,0,0,0,0.1,0,0,0,0,0)	5e-05	4e-05
(0,0,0,0,0,0,0,30,0,0)	(0,0,0,0,0,0,0,0.8,0,0)	0.55065	0.60173
(0,0,0,0,0,0,0,30,0,0)	(0,0,0,0,0,0,0,0.6,0,0)	0.07225	0.07358
(0,0,0,0,0,0,0,30,0,0)	(0,0,0,0,0,0,0,0.5,0,0)	0.07225	0.00375
(0,0,0,0,0,0,0,0,0,30)	(0,0,0,0,0,0,0,0,0,0)	0	0
(10,0,0,0,0,0,0,0,0,0)	(1,0,0,0,0,0,0,0,0,0)	0	0

Table 14: Hanea's and Nane's calibration score and calibration score bin 4 for given binassignment using 30 questions.

Bin assignment	s_i	Bin	Bin 2
(0,0,0,0,0,0,0,0,0,50)	(0,0,0,0,0,0,0,0,0,1)	0.07694	0
(0,0,0,0,50,0,0,0,0,0)	(0,0,0,0,0.5,0,0,0,0,0)	0.48059	0.39326
(0,0,50,0,0,0,0,0,0,0)	(0,0,0.3,0,0,0,0,0,0,0)	0.415	0.32617
(0,0,0,0,50,0,0,0,0,0)	(0,0,0,0,0.3,0,0,0,0,0)	0.03233	0.02077
(0,0,0,0,50,0,0,0,0,0)	(0,0,0,0,0.2,0,0,0,0,0)	0.00026	0.00011
(0,0,0,0,50,0,0,0,0,0)	(0,0,0,0,0.1,0,0,0,0,0)	0	0
(0,0,0,0,0,0,0,50,0,0)	(0,0,0,0,0,0,0,0.8,0,0)	0.42589	0.32737
(0,0,0,0,0,0,0,50,0,0)	(0,0,0,0,0,0,0,0.6,0,0)	0.02018	0.01253
(0,0,0,0,0,0,0,50,0,0)	(0,0,0,0,0,0,0,0.5,0,0)	0.00016	8e-05
(0,0,0,0,0,0,0,0,0,50)	(0,0,0,0,0,0,0,0,0,0)	0	0
(50,0,0,0,0,0,0,0,0,0)	(1,0,0,0,0,0,0,0,0,0)	0	0

Table 15: Hanea's and Nane's calibration score and calibration score bin 2 for given bin assignment using 50 questions.

Bin assignment	s_i	Bin	Bin 3
(0,0,0,0,0,0,0,0,0,50)	(0,0,0,0,0,0,0,0,0,1)	0.07694	0.18056
(0,0,0,0,50,0,0,0,0,0)	(0,0,0,0,0.5,0,0,0,0,0)	0.48059	0.48133
(0,0,50,0,0,0,0,0,0,0)	(0,0,0.3,0,0,0,0,0,0,0)	0.415	0.4156
(0,0,0,0,50,0,0,0,0,0)	(0,0,0,0,0.3,0,0,0,0,0)	0.03233	0.03355
(0,0,0,0,50,0,0,0,0,0)	(0,0,0,0,0.2,0,0,0,0,0)	0.00026	3e-04
(0,0,0,0,50,0,0,0,0,0)	(0,0,0,0,0.1,0,0,0,0,0)	0	0
(0,0,0,0,0,0,0,50,0,0)	(0,0,0,0,0,0,0,0.8,0,0)	0.42589	0.51412
(0,0,0,0,0,0,0,50,0,0)	(0,0,0,0,0,0,0,0.6,0,0)	0.02018	0.02096
(0,0,0,0,0,0,0,50,0,0)	(0,0,0,0,0,0,0,0.5,0,0)	0.00016	0.00021
(0,0,0,0,0,0,0,0,0,50)	(0,0,0,0,0,0,0,0,0,0)	0	0
(50,0,0,0,0,0,0,0,0,0)	(1,0,0,0,0,0,0,0,0,0)	0	0

Table 16: Hanea's and Nane's calibration score and calibration score bin 3 for given bin assignment using 50 questions.

Bin assignment	s_i	Bin	Bin 4
(0,0,0,0,0,0,0,0,0,50)	(0,0,0,0,0,0,0,0,0,1)	0.07694	0.14209
(0,0,0,0,50,0,0,0,0,0)	(0,0,0,0,0.5,0,0,0,0,0)	0.48059	0.43766
(0,0,50,0,0,0,0,0,0,0)	(0,0,0.3,0,0,0,0,0,0,0)	0.415	0.37118
(0,0,0,0,50,0,0,0,0,0)	(0,0,0,0,0.3,0,0,0,0,0)	0.03233	0.02777
(0,0,0,0,50,0,0,0,0,0)	(0,0,0,0,0.2,0,0,0,0,0)	0.00026	0.00023
(0,0,0,0,50,0,0,0,0,0)	(0,0,0,0,0.1,0,0,0,0,0)	0	0
(0,0,0,0,0,0,0,50,0,0)	(0,0,0,0,0,0,0,0.8,0,0)	0.42589	0.46486
(0,0,0,0,0,0,0,50,0,0)	(0,0,0,0,0,0,0,0.6,0,0)	0.02018	0.01714
(0,0,0,0,0,0,0,50,0,0)	(0,0,0,0,0,0,0,0.5,0,0)	0.00016	0.00017
(0,0,0,0,0,0,0,0,0,50)	(0,0,0,0,0,0,0,0,0,0)	0	0
(50,0,0,0,0,0,0,0,0,0)	(1,0,0,0,0,0,0,0,0,0)	0	0

Table 17: Hanea's and Nane's calibration score and calibration score bin 4 for given bin assignment using 50 questions.

Bin assignment	s_i	Bin	Bin 2
(0,0,0,0,0,0,0,0,0,100)	(0,0,0,0,0,0,0,0,0,1)	0.00592	0
(0,0,0,0,100,0,0,0,0,0)	(0,0,0,0,0.5,0,0,0,0,0)	0.3173	0.26915
(0,0,100,0,0,0,0,0,0,0)	(0,0,0.3,0,0,0,0,0,0,0)	0.25333	0.20757
(0,0,0,0,100,0,0,0,0,0)	(0,0,0,0,0.3,0,0,0,0,0)	0.0023	0.20757
(0,0,0,0,100,0,0,0,0,0)	(0,0,0,0,0.2,0,0,0,0,0)	0	0
(0,0,0,0,100,0,0,0,0,0)	(0,0,0,0,0.1,0,0,0,0,0)	0	0
(0,0,0,0,0,0,0,100,0,0)	(0,0,0,0,0,0,0,0.8,0,0)	0.24836	0.19906
(0,0,0,0,0,0,0,100,0,0)	(0,0,0,0,0,0,0,0.6,0,0)	0.00101	0.00065
(0,0,0,0,0,0,0,100,0,0)	(0,0,0,0,0,0,0,0.5,0,0)	0	0
(0,0,0,0,0,0,0,0,0,100)	(0,0,0,0,0,0,0,0,0,0)	0	0
(100,0,0,0,0,0,0,0,0,0)	(1,0,0,0,0,0,0,0,0,0)	0	0

Table 18: Hanea's and Nane's calibration score and calibration score bin 2 for given bin assignment using 100 questions.

Bin assignment	s_i	Bin	Bin 3
(0,0,0,0,0,0,0,0,0,100)	(0,0,0,0,0,0,0,0,0,1)	0.00592	0.01019
(0,0,0,0,100,0,0,0,0,0)	(0,0,0,0,0.5,0,0,0,0,0)	0.3173	0.31698
(0,0,100,0,0,0,0,0,0,0)	(0,0,0.3,0,0,0,0,0,0,0)	0.25333	0.24907
(0,0,0,0,100,0,0,0,0,0)	(0,0,0,0,0.3,0,0,0,0,0)	0.0023	0.00247
(0,0,0,0,100,0,0,0,0,0)	(0,0,0,0,0.2,0,0,0,0,0)	0	0
(0,0,0,0,100,0,0,0,0,0)	(0,0,0,0,0.1,0,0,0,0,0)	0	0
(0,0,0,0,0,0,0,100,0,0)	(0,0,0,0,0,0,0,0.8,0,0)	0.24836	0.29837
(0,0,0,0,0,0,0,100,0,0)	(0,0,0,0,0,0,0,0.6,0,0)	0.00101	0.00108
(0,0,0,0,0,0,0,100,0,0)	(0,0,0,0,0,0,0,0.5,0,0)	0	0
(0,0,0,0,0,0,0,0,0,100)	(0,0,0,0,0,0,0,0,0,0)	0	0
(100,0,0,0,0,0,0,0,0,0)	(1,0,0,0,0,0,0,0,0,0)	0	0

Table 19: Hanea's and Nane's calibration score and calibration score bin 3 for given bin assignment using 100 questions.

Bin assignment	s_i	Bin	Bin 4
(0,0,0,0,0,0,0,0,0,100)	(0,0,0,0,0,0,0,0,0,1)	0.00592	0.00723
(0,0,0,0,100,0,0,0,0,0)	(0,0,0,0,0.5,0,0,0,0,0)	0.3173	0.29291
(0,0,100,0,0,0,0,0,0,0)	(0,0,0.3,0,0,0,0,0,0,0)	0.25333	0.22619
(0,0,0,0,100,0,0,0,0,0)	(0,0,0,0,0.3,0,0,0,0,0)	0.0023	0.00209
(0,0,0,0,100,0,0,0,0,0)	(0,0,0,0,0.2,0,0,0,0,0)	0	0
(0,0,0,0,100,0,0,0,0,0)	(0,0,0,0,0.1,0,0,0,0,0)	0	0
(0,0,0,0,0,0,0,100,0,0)	(0,0,0,0,0,0,0,0.8,0,0)	0.24836	0.27372
(0,0,0,0,0,0,0,100,0,0)	(0,0,0,0,0,0,0,0.6,0,0)	0.00101	9e-04
(0,0,0,0,0,0,0,100,0,0)	(0,0,0,0,0,0,0,0.5,0,0)	0	0
(0,0,0,0,0,0,0,0,0,100)	(0,0,0,0,0,0,0,0,0,0)	0	0
(100,0,0,0,0,0,0,0,0,0)	(1,0,0,0,0,0,0,0,0,0)	0	0

Table 20: Hanea's and Nane's calibration score and calibration score bin 4 for given bin assignment using 100 questions.

B R code for section 3

This code was used for (Hanea and Nane, 2019) and during this project small adjustments and additions have been made.

```
#define the theoretical p_b
p_i<-seq(5,95,by=10)/100

#cal score based on the chi square dist
#the function computes for each expert (with index x) the calibration scores
calibration_chi_simulations<-function(s_i,n_i)
{
  p1<-s_i*log(s_i/p_i)
  p1[is.nan(p1)]<-0
  p2<-(1-s_i)*log((1-s_i)/(1-p_i))
  p2[is.nan(p2)]<-0

  rel_info<-p1+p2
  sum_rel_info<-sum(2*n_i*rel_info)
  calibration<-1-pchisq(sum_rel_info,10)#10 bins so 10 degrees of freedom
  return(calibration)
}

#simulation results
calibration_chi_simulations(c(0,0,0,0,0,0,0,0,0,1), c(0,0,0,0,0,0,0,0,0,10))

calibration_chi_simulations(c(0,0,0,0,0.5,0,0,0,0,0), c(0,0,0,0,10,0,0,0,0,0))

calibration_chi_simulations(c(0,0,0.3,0,0,0,0,0,0,0), c(0,0,10,0,0,0,0,0,0,0))

calibration_chi_simulations(c(0,0,0,0,0.3,0,0,0,0,0), c(0,0,0,0,10,0,0,0,0,0))

calibration_chi_simulations(c(0,0,0,0,0.2,0,0,0,0,0), c(0,0,0,0,10,0,0,0,0,0))

calibration_chi_simulations(c(0,0,0,0,0.1,0,0,0,0,0), c(0,0,0,0,10,0,0,0,0,0))

calibration_chi_simulations(c(0,0,0,0,0,0,0,0.8,0,0), c(0,0,0,0,0,0,0,10,0,0))
```

```
calibration_chi_simulations(c(0,0,0,0,0,0,0,0.6,0,0), c(0,0,0,0,0,0,0,10,0,0))

calibration_chi_simulations(c(0,0,0,0,0,0,0,0.5,0,0), c(0,0,0,0,0,0,0,10,0,0))

#test

calibration_chi_simulations(c(0,0,0,0,0.1,0,0,0,0,0), c(0,0,0,0,30,0,0,0,0,0))

calibration_chi_simulations(c(0,0,0,0,0,0,0,0,0,1), c(0,0,0,0,0,0,0,0,0,30))


calibration_bin_simulations<-function(s_i,n_i){
  sum_n_i_s_i<-sum(n_i*s_i)

  x<-get.exact.binsum(n_i,p_i)
  x<-rbind(c(-1,0,0),x)

  k=which(x[,1]==sum_n_i_s_i)
  mid_p_value<-2*min(1-x[k,3]+1/2*x[k,2], x[k-1,3]+1/2*x[k,2])
  return(round(mid_p_value,5))
}

calibration_bin_simulations(c(0,0,0,0,0,0,0,0,0,1), c(0,0,0,0,0,0,0,0,0,10))

calibration_bin_simulations(c(0,0,0,0,0.5,0,0,0,0,0), c(0,0,0,0,10,0,0,0,0,0))

calibration_bin_simulations(c(0,0,0.3,0,0,0,0,0,0,0), c(0,0,10,0,0,0,0,0,0,0))

calibration_bin_simulations(c(0,0,0,0,0.3,0,0,0,0,0), c(0,0,0,0,10,0,0,0,0,0))

calibration_bin_simulations(c(0,0,0,0,0.2,0,0,0,0,0), c(0,0,0,0,10,0,0,0,0,0))

calibration_bin_simulations(c(0,0,0,0,0.1,0,0,0,0,0), c(0,0,0,0,10,0,0,0,0,0))

calibration_bin_simulations(c(0,0,0,0,0,0,0,0.8,0,0), c(0,0,0,0,0,0,0,10,0,0))
```

```
calibration_bin_simulations(c(0,0,0,0,0,0,0,0.6,0,0), c(0,0,0,0,0,0,0,10,0,0))

calibration_bin_simulations(c(0,0,0,0,0,0,0,0.5,0,0), c(0,0,0,0,0,0,0,10,0,0))

#calibration score based on first two-sided test formula from Rivals et Al
calibration_bin_simulations2<-function(s_i,n_i){
  sum_n_i_s_i<-sum(n_i*s_i)

  x<-get.exact.binsum(n_i,p_i)
  x<-rbind(c(-1,0,0),x)

  k=which(x[,1]==sum_n_i_s_i)
  twice_onesided_p_value = 2*min(1-x[k,3], x[k-1,3])
  return(round(twice_onesided_p_value,5))
}

calibration_bin_simulations2(c(0,0,0,0,0,0,0,0,0,1), c(0,0,0,0,0,0,0,0,0,10))

calibration_bin_simulations2(c(0,0,0,0,0.5,0,0,0,0,0), c(0,0,0,0,10,0,0,0,0,0))

calibration_bin_simulations2(c(0,0,0.3,0,0,0,0,0,0,0), c(0,0,10,0,0,0,0,0,0,0))

calibration_bin_simulations2(c(0,0,0,0,0.3,0,0,0,0,0), c(0,0,0,0,10,0,0,0,0,0))

calibration_bin_simulations2(c(0,0,0,0,0.2,0,0,0,0,0), c(0,0,0,0,10,0,0,0,0,0))

calibration_bin_simulations2(c(0,0,0,0,0.1,0,0,0,0,0), c(0,0,0,0,10,0,0,0,0,0))

calibration_bin_simulations2(c(0,0,0,0,0,0,0,0.8,0,0), c(0,0,0,0,0,0,0,10,0,0))

calibration_bin_simulations2(c(0,0,0,0,0,0,0,0.6,0,0), c(0,0,0,0,0,0,0,10,0,0))

calibration_bin_simulations2(c(0,0,0,0,0,0,0,0.5,0,0), c(0,0,0,0,0,0,0,10,0,0))

calibration_bin_simulations3<-function(s_i,n_i){
  sum_n_i_s_i<-sum(n_i*s_i)
```

```
  x<-get.exact.binsum(n_i,p_i)
  x<-rbind(c(-1,0,0),x)
  k=which(x[,1]==sum_n_i_s_i)
  min_lik_p_value = 0
  for (j in 1:nrow(x)){if (x[j,2]<=x[k,2]){min_lik_p_value=min_lik_p_value+x[j,2

  return(round(min_lik_p_value,5))
}

calibration_bin_simulations3(c(0,0,0,0,0,0,0,0,0,1), c(0,0,0,0,0,0,0,0,0,10))

calibration_bin_simulations3(c(0,0,0,0,0.5,0,0,0,0,0), c(0,0,0,0,10,0,0,0,0,0))

calibration_bin_simulations3(c(0,0,0.3,0,0,0,0,0,0,0), c(0,0,10,0,0,0,0,0,0,0))

calibration_bin_simulations3(c(0,0,0,0,0.3,0,0,0,0,0), c(0,0,0,0,10,0,0,0,0,0))

calibration_bin_simulations3(c(0,0,0,0,0.2,0,0,0,0,0), c(0,0,0,0,10,0,0,0,0,0))

calibration_bin_simulations3(c(0,0,0,0,0.1,0,0,0,0,0), c(0,0,0,0,10,0,0,0,0,0))

calibration_bin_simulations3(c(0,0,0,0,0,0,0,0.8,0,0), c(0,0,0,0,0,0,0,10,0,0))

calibration_bin_simulations3(c(0,0,0,0,0,0,0,0.6,0,0), c(0,0,0,0,0,0,0,10,0,0))

calibration_bin_simulations3(c(0,0,0,0,0,0,0,0.5,0,0), c(0,0,0,0,0,0,0,10,0,0))

#test
calibration_bin_simulations(s1, n1)

calibration_bin_simulations4<-function(s_i,n_i){
  sum_n_i_s_i<-sum(n_i*s_i)

  x<-get.exact.binsum(n_i,p_i)
  x<-rbind(c(-1,0,0),x)
```

```
  k=which(x[,1]==sum_n_i_s_i)
  min_lik_pi_value = 0
  for (j in 1:nrow(x)){
    if (x[j,2]<x[k,2]){min_lik_pi_value=min_lik_pi_value+x[j,2]}
    else if(x[j,2]==x[k,2])
    {min_lik_pi_value=min_lik_pi_value+0.5*x[j,2]}
    }

  return(round(min_lik_pi_value,5))
}

calibration_bin_simulations4(c(0,0,0,0,0,0,0,0,0,1), c(0,0,0,0,0,0,0,0,0,10))

calibration_bin_simulations4(c(0,0,0,0,0.5,0,0,0,0,0), c(0,0,0,0,10,0,0,0,0,0))

calibration_bin_simulations4(c(0,0,0.3,0,0,0,0,0,0,0), c(0,0,10,0,0,0,0,0,0,0))

calibration_bin_simulations4(c(0,0,0,0,0.3,0,0,0,0,0), c(0,0,0,0,10,0,0,0,0,0))

calibration_bin_simulations4(c(0,0,0,0,0.2,0,0,0,0,0), c(0,0,0,0,10,0,0,0,0,0))

calibration_bin_simulations4(c(0,0,0,0,0.1,0,0,0,0,0), c(0,0,0,0,10,0,0,0,0,0))

calibration_bin_simulations4(c(0,0,0,0,0,0,0,0.8,0,0), c(0,0,0,0,0,0,0,0,10,0,0))

calibration_bin_simulations4(c(0,0,0,0,0,0,0,0.6,0,0), c(0,0,0,0,0,0,0,0,10,0,0))

calibration_bin_simulations4(c(0,0,0,0,0,0,0,0.5,0,0), c(0,0,0,0,0,0,0,0,10,0,0))

# for bin

n1 = c(0,0,0,0,0,0,0,0,0,20)
n2 = c(0,0,0,0,0,0,0,0,0,30)
n3 = c(0,0,0,0,30,0,0,0,0,0)
n4 = c(0,0,30,0,0,0,0,0,0,0)
n5 = c(0,0,0,0,30,0,0,0,0,0)
```

```
n6 = c(0,0,0,0,30,0,0,0,0,0)
n7 = c(0,0,0,0,30,0,0,0,0,0)
n8 = c(0,0,0,0,0,0,0,30,0,0)
n9 = c(0,0,0,0,0,0,0,30,0,0)
n10 = c(0,0,0,0,0,0,0,30,0,0)

s1 = c(0,0,0,0,0,0,0,0,0,1)
s2 = c(0,0,0,0,0,0,0,0,0,1)
s3 = c(0,0,0,0,0.5,0,0,0,0,0)
s4 = c(0,0,0.3,0,0,0,0,0,0,0)
s5 = c(0,0,0,0,0.3,0,0,0,0,0)
s6 = c(0,0,0,0,0.2,0,0,0,0,0)
s7 = c(0,0,0,0,0.1,0,0,0,0,0)
s8 = c(0,0,0,0,0,0,0,0.8,0,0)
s9 = c(0,0,0,0,0,0,0,0.6,0,0)
s10 = c(0,0,0,0,0,0,0,0.5,0,0)

n1s1 = n1*s1
n2s2 = n2*s2
n3s3 = n3*s3
n4s4 = n4*s4
n5s5 = n5*s5
n6s6 = n6*s6
n7s7 = n7*s7
n8s8 = n8*s8
n9s9 = n9*s9
n10s10 = n10*s10

sum_n1_s1 = sum(n1s1)
sum_n2_s2 = sum(n2s2)
sum_n3_s3 = sum(n3s3)
sum_n4_s4 = sum(n4s4)
sum_n5_s5 = sum(n5s5)
sum_n6_s6 = sum(n6s6)
sum_n7_s7 = sum(n7s7)
sum_n8_s8 = sum(n8s8)
```

```
sum_n9_s9 = sum(n9s9)
sum_n10_s10 = sum(n10s10)

x1<-get.exact.binsum(n1, p_i)
x1<-rbind(c(-1,0,0),x1)
k1=which(x1[,1]==sum_n1_s1)
1-x1[k1,3]+1/2*x1[k1,2]
x1[k1-1,3]+1/2*x1[k1,2]


x2<-get.exact.binsum(n2, p_i)
x2<-rbind(c(-1,0,0),x2)
k2=which(x2[,1]==sum_n2_s2)
1-x2[k2,3]+1/2*x2[k2,2]
x2[k2-1,3]+1/2*x2[k2,2]

x3<-get.exact.binsum(n3, s3)
x3<-rbind(c(-1,0,0),x3)
k3=which(x3[,1]==sum_n3_s3)
1-x3[k3,3]+1/2*x3[k3,2]
x3[k3-1,3]+1/2*x3[k3,2]

x4<-get.exact.binsum(n4, s4)
x4<-rbind(c(-1,0,0),x4)
k4=which(x4[,1]==sum_n4_s4)
1-x4[k4,3]+1/2*x4[k4,2]
x4[k1-1,3]+1/2*x4[k1,2]

x5<-get.exact.binsum(n5, s5)
x5<-rbind(c(-1,0,0),x5)
k5=which(x5[,1]==sum_n5_s5)
1-x5[k5,3]+1/2*x5[k5,2]
x5[k5-1,3]+1/2*x5[k5,2]

x6<-get.exact.binsum(n6, s6)
x6<-rbind(c(-1,0,0),x6)
```

```
k6=which(x7[,1]==sum_n6_s6)
1-x6[k6,3]+1/2*x6[k6,2]
x6[k6-1,3]+1/2*x6[k6,2]

x7<-get.exact.binsum(n7,s7)
x7<-rbind(c(-1,0,0),x7)
k7=which(x7[,1]==sum_n7_s7)
1-x7[k7,3]+1/2*x7[k7,2]
x7[k7-1,3]+1/2*x7[k7,2]

x8<-get.exact.binsum(n8,s8)
x8<-rbind(c(-1,0,0),x8)
k8=which(x8[,1]==sum_n8_s8)
1-x8[k8,3]+1/2*x8[k8,2]
x8[k8-1,3]+1/2*x8[k8,2]

x9<-get.exact.binsum(n9,s9)
x9<-rbind(c(-1,0,0),x9)
k9=which(x9[,1]==sum_n9_s9)
1-x9[k9,3]+1/2*x9[k9,2]
x9[k9-1,3]+1/2*x9[k9,2]

x10<-get.exact.binsum(n10,s10)
x10<-rbind(c(-1,0,0),x10)
k10=which(x10[,1]==sum_n10_s10)
1-x10[k10,3]+1/2*x10[k10,2]
x10[k10-1,3]+1/2*x10[k10,2]

# for chi

p1_1<-s1*log(s1/p_i)
p1_1[is.nan(p1_1)]<-0
p2_1<-(1-s1)*log((1-s1)/(1-p_i))
p2_1[is.nan(p2_1)]<-0
rel_info1<-p1_1+p2_1
sum_rel_info1<-sum(2*n1*rel_info1)
```

```
calibration1<-1-pchisq(sum_rel_info1,10)

p1_2<-s1*log(s2/p_i)
p1_2[is.nan(p1_2)]<-0
p2_2<-(1-s2)*log((1-s2)/(1-p_i))
p2_2[is.nan(p2_2)]<-0
rel_info2<-p1_2+p2_2
sum_rel_info2<-sum(2*n2*rel_info2)
calibration2<-1-pchisq(sum_rel_info2,10)

p1_3<-s3*log(s3/p_i)
p1_3[is.nan(p1_3)]<-0
p2_3<-(1-s3)*log((1-s3)/(1-p_i))
p2_3[is.nan(p2_3)]<-0
rel_info3<-p1_3+p2_3
sum_rel_info3<-sum(2*n3*rel_info3)
calibration3<-1-pchisq(sum_rel_info3,10)

p1_4<-s4*log(s1/p_i)
p1_4[is.nan(p1_4)]<-0
p2_4<-(1-s4)*log((1-s4)/(1-p_i))
p2_4[is.nan(p2_4)]<-0
rel_info4<-p1_4+p2_4
sum_rel_info4<-sum(2*n4*rel_info4)
calibration4<-1-pchisq(sum_rel_info4,10)

p1_5<-s5*log(s5/p_i)
p1_5[is.nan(p1_5)]<-0
p2_5<-(1-s5)*log((1-s5)/(1-p_i))
p2_5[is.nan(p2_5)]<-0
rel_info5<-p1_5+p2_5
sum_rel_info5<-sum(2*n5*rel_info5)
calibration5<-1-pchisq(sum_rel_info5,10)

p1_6<-s6*log(s6/p_i)
p1_6[is.nan(p1_6)]<-0
```

```
p2_6<-(1-s6)*log((1-s6)/(1-p_i))
p2_6[is.nan(p2_6)]<-0
rel_info6<-p1_6+p2_6
sum_rel_info6<-sum(2*n6*rel_info6)
calibration6<-1-pchisq(sum_rel_info6,10)

p1_7<-s7*log(s7/p_i)
p1_7[is.nan(p1_7)]<-0
p2_7<-(1-s7)*log((1-s7)/(1-p_i))
p2_7[is.nan(p2_7)]<-0
rel_info7<-p1_7+p2_7
sum_rel_info7<-sum(2*n7*rel_info7)
calibration7<-1-pchisq(sum_rel_info7,10)

p1_8<-s8*log(s8/p_i)
p1_8[is.nan(p1_8)]<-0
p2_8<-(1-s8)*log((1-s8)/(1-p_i))
p2_8[is.nan(p2_8)]<-0
rel_info8<-p1_8+p2_8
sum_rel_info8<-sum(2*n8*rel_info8)
calibration8<-1-pchisq(sum_rel_info8,10)

p1_9<-s9*log(s9/p_i)
p1_9[is.nan(p1_9)]<-0
p2_9<-(1-s9)*log((1-s9)/(1-p_i))
p2_9[is.nan(p2_9)]<-0
rel_info9<-p1_9+p2_9
sum_rel_info9<-sum(2*n9*rel_info9)
calibration9<-1-pchisq(sum_rel_info9,10)

p1_10<-s9*log(s10/p_i)
p1_10[is.nan(p1_10)]<-0
p2_10<-(1-s10)*log((1-s10)/(1-p_i))
p2_10[is.nan(p2_10)]<-0
rel_info10<-p1_10+p2_10
sum_rel_info10<-sum(2*n10*rel_info10)
```

```
calibration10<-1-pchisq(sum_rel_info10,10)
```

C R code for section 4

This code was used for (Hanea and Nane, 2019) and during this project small adjustments and additions have been made.

```
require(openxlsx)
library(ggplot2)
require(grid)
require(gridExtra)
require(cowplot)
##################################
#####import data###############
##################################


#dataIG4years is data imported from dataIG4years file
#dataACE_GJP is data imported from dataACE_GJP file
#dataACE_IDEA is data imported from dataACE_IDEA file
#dataSIPS is data imported from dataSIPS file

#need to define those for the computations
experts_IG4years<-dataIG4years[,1]
questions_IG4years<-dataIG4years[,2]
answers_IG4years<-dataIG4years[,7]
realizations_IG4years<-dataIG4years[,9]

#need to define those for the computations
experts_dataACE_GJP<-dataACE_GJP[,1]
questions_dataACE_GJP<-dataACE_GJP[,2]
answers_dataACE_GJP<-dataACE_GJP[,3]
realizations_dataACE_GJP<-dataACE_GJP[,4]

#need to define those for the computations
experts_dataACE_IDEA<-dataACE_IDEA[,1]
questions_dataACE_IDEA<-dataACE_IDEA[,2]
answers_dataACE_IDEA<-dataACE_IDEA[,7]
realizations_dataACE_IDEA<-dataACE_IDEA[,9]
```

```
#need to define those for the computations
experts_dataSIPS<-dataSIPS[,1]
questions_dataSIPS<-dataSIPS[,2]
answers_dataSIPS<-dataSIPS[,7]
realizations_dataSIPS<-dataSIPS[,9]

unique_questions_IG4years<-(unique(questions_IG4years))
unique_experts_IG4years<-(unique(experts_IG4years))

unique_questions_dataACE_GJP<-(unique(questions_dataACE_GJP))
unique_expert_dataACE_GJP<-(unique(experts_dataACE_GJP))

unique_questions_dataACE_IDEA<-(unique(questions_dataACE_IDEA))
unique_experts_dataACE_IDEA<-(unique(experts_dataACE_IDEA))

unique_questions_dataSIPS<-(unique(questions_dataSIPS))
unique_experts_dataSIPS<-(unique(experts_dataSIPS))

#################################
###calibration scores#########
#################################

p<-seq(5,95,by=10)/100
p_i<-seq(5,95,by=10)/100

#function to assign experts' assessments to bins
#the function provides results per expert, and x stands for index of
expert
exp_to_bin<- function(experts,questions,answers,x)
{
  unique_experts<-(unique(experts))
  ans<-answers[which(experts==unique_experts[x])]
  ifelse (ans==1, expert_answers<-10, expert_answers<-ceiling((ans+10^(-6))*10))

  return(expert_answers)
```

```
}

#function which displays realizations of events for given expert
outcomes = function(experts, realizations, x)
{
  unique_experts=(unique(experts))
  outcome = realizations[which(experts==unique_experts[x])]
  return(outcome)
}

#some tests
exp_to_bin(experts_dataACE_GJP,questions_dataACE_GJP, answers_dataACE_GJP,1)
exp_to_bin(experts_dataACE_IDEA,questions_dataACE_IDEA, answers_dataACE_IDEA,1)
exp_to_bin(experts_dataACE_IDEA,questions_dataACE_IDEA, answers_dataACE_SIPS,1)

outcomes(experts_dataACE_GJP, realizations_dataACE_GJP,1)

#function to compute n
compute_n_i = function(binassign)
{
  b<-1:10
  n_i<-sapply(1:10, function(x) length(which(binassign%in%b[x])))
  return(n_i)
}

#function to compute s
compute_s_i = function(binassign, outcome)
{
  a = binassign
  b = 1:10
  s_i<-sapply(1:10, function(x)
  sum(outcome[which(a%in%b[x])])/length(outcome[which(a%in%b[x])]))
  #be careful with the NaN values
  s_i[is.nan(s_i)]<-0
  return(s_i)
}
```

```
##############################################
#cal score based on the chi square dist
#the function computes for each expert (with index x) the calibration scores
calibration_chi<-function(s_i, n_i)
{
  p1<-s_i*log(s_i/p)
  p1[is.nan(p1)]<-0
  p2<-(1-s_i)*log((1-s_i)/(1-p))
  p2[is.nan(p2)]<-0

  rel_info<-p1+p2
  sum_rel_info<-sum(2*n_i*rel_info)
  calibration<-1-pchisq(sum_rel_info,9)
  return(calibration)
}

##cal score based on the exact distribution and the mid p-value
##Binomial approach of computing calibration scores with p-values

calibration_bin_simulations<-function(s_i,n_i){
  sum_n_i_s_i<-sum(n_i*s_i)

  x<-get.exact.binsum(n_i,p_i)
  x<-rbind(c(-1,0,0),x)

  k=which(x[,1]==sum_n_i_s_i)
  mid_p_value<-2*min(1-x[k,3]+1/2*x[k,2], x[k-1,3]+1/2*x[k,2])

  return(round(mid_p_value,5))
}

#calibration score based on first two-sided test formula from Rivals et Al
calibration_bin_simulations2<-function(s_i,n_i){
  sum_n_i_s_i<-sum(n_i*s_i)
```

```
  x<-get.exact.binsum(n_i,p_i)
  x<-rbind(c(-1,0,0),x)
  k=which(x[,1]==sum_n_i_s_i)
  twice_onesided_p_value = 2*min(1-x[k,3], x[k-1,3])

  return(round(twice_onesided_p_value,5))
}

calibration_bin_simulations3<-function(s_i,n_i){
  sum_n_i_s_i<-sum(n_i*s_i)

  x<-get.exact.binsum(n_i,p_i)
  x<-rbind(c(-1,0,0),x)

  k=which(x[,1]==sum_n_i_s_i)
  min_lik_p_value = 0
  for (j in 1:nrow(x)){if (x[j,2]<=x[k,2]){min_lik_p_value=min_lik_p_value+
  x[j,2]}}

  return(round(min_lik_p_value,5))
}

calibration_bin_simulations4<-function(s_i,n_i){
  sum_n_i_s_i<-sum(n_i*s_i)

  x<-get.exact.binsum(n_i,p_i)
  x<-rbind(c(-1,0,0),x)
  k=which(x[,1]==sum_n_i_s_i)
  min_lik_pi_value = 0
  for (j in 1:nrow(x)){
    if (x[j,2]<x[k,2]){min_lik_pi_value=min_lik_pi_value+x[j,2]}
    else if(x[j,2]==x[k,2])
    {min_lik_pi_value=min_lik_pi_value+0.5*x[j,2]}
  }
    return(round(min_lik_pi_value,5))
}
```

```
cal_scores_chi_unique_dataACE_GJP = c()
for (j in 1:length(unique_expert_dataACE_GJP))
{
  n = compute_n_i(exp_to_bin(experts_dataACE_GJP,questions_dataACE_GJP,
  answers_dataACE_GJP,j))
  s = compute_s_i(exp_to_bin(experts_dataACE_GJP,questions_dataACE_GJP,
  answers_dataACE_GJP,j), outcomes(experts_dataACE_GJP,
  realizations_dataACE_GJP,j))
  cal_scores_chi_unique_dataACE_GJP = append(cal_scores_chi_unique_dataACE_GJP,
  calibration_chi(s, n))
}

cal_scores_bin_unique_dataACE_GJP = c()
for (j in 1:length(unique_expert_dataACE_GJP))
{
  n = compute_n_i(exp_to_bin(experts_dataACE_GJP,questions_dataACE_GJP,
  answers_dataACE_GJP,j))
  s = compute_s_i(exp_to_bin(experts_dataACE_GJP,questions_dataACE_GJP,
  answers_dataACE_GJP,j), outcomes(experts_dataACE_GJP,
  realizations_dataACE_GJP,j))
  cal_scores_bin_unique_dataACE_GJP = append(cal_scores_bin_unique_dataACE_GJP,
  calibration_bin_simulations(s, n))
}

cal_scores_bin2_unique_dataACE_GJP = c()
for (j in 1:length(unique_expert_dataACE_GJP))
{
  n = compute_n_i(exp_to_bin(experts_dataACE_GJP,questions_dataACE_GJP,
  answers_dataACE_GJP,j))
  s = compute_s_i(exp_to_bin(experts_dataACE_GJP,questions_dataACE_GJP,
  answers_dataACE_GJP,j), outcomes(experts_dataACE_GJP,
  realizations_dataACE_GJP,j))
  cal_scores_bin2_unique_dataACE_GJP =
  append(cal_scores_bin2_unique_dataACE_GJP, calibration_bin_simulations2(s, n))
}
```

```
cal_scores_bin3_unique_dataACE_GJP = c()
for (j in 1:length(unique_expert_dataACE_GJP))
{
  n = compute_n_i(exp_to_bin(experts_dataACE_GJP,questions_dataACE_GJP,
  answers_dataACE_GJP,j))
  s = compute_s_i(exp_to_bin(experts_dataACE_GJP,questions_dataACE_GJP,
  answers_dataACE_GJP,j), outcomes(experts_dataACE_GJP,
  realizations_dataACE_GJP,j))
  cal_scores_bin3_unique_dataACE_GJP =
  append(cal_scores_bin3_unique_dataACE_GJP, calibration_bin_simulations3(s, n))
}

cal_scores_bin4_unique_dataACE_GJP = c()
for (j in 1:length(unique_expert_dataACE_GJP))
{
  n = compute_n_i(exp_to_bin(experts_dataACE_GJP,questions_dataACE_GJP,
  answers_dataACE_GJP,j))
  s = compute_s_i(exp_to_bin(experts_dataACE_GJP,questions_dataACE_GJP,
  answers_dataACE_GJP,j), outcomes(experts_dataACE_GJP,
  realizations_dataACE_GJP,j))
  cal_scores_bin4_unique_dataACE_GJP =
  append(cal_scores_bin4_unique_dataACE_GJP, calibration_bin_simulations4(s, n))
}

data <- data.frame(
  expert = unique_expert_dataACE_GJP,
  chi = cal_scores_chi_unique_dataACE_GJP,
  bin = cal_scores_bin_unique_dataACE_GJP,
  bin2 = cal_scores_bin2_unique_dataACE_GJP,
  bin3 = cal_scores_bin3_unique_dataACE_GJP,
  bin4 = cal_scores_bin4_unique_dataACE_GJP
)

gjp_from_than_ten_upto_100_questions
=calibr_scores_dataACE_GJP_file
```

```
[calibr_scores_dataACE_GJP_file$exp_questions_dataACE_GJP > 9 &
calibr_scores_dataACE_GJP_file$exp_questions_dataACE_GJP < 101,]
gjp_more_than_100_questions=calibr_scores_dataACE_GJP_file
[calibr_scores_dataACE_GJP_file$exp_questions_dataACE_GJP > 100,]

exp_questions_dataACE_GJP<-sapply(1:length(unique_expert_dataACE_GJP),
function(x) length(which(experts_dataACE_GJP==(unique_expert_dataACE_GJP[x]))))

calibr_scores_dataACE_GJP<-cbind(unique_expert_dataACE_GJP,
exp_questions_dataACE_GJP, cal_scores_chi_unique_dataACE_GJP,
cal_scores_bin_unique_dataACE_GJP, cal_scores_bin2_unique_dataACE_GJP,
cal_scores_bin3_unique_dataACE_GJP, cal_scores_bin4_unique_dataACE_GJP)

hist(data$chi, ylab="Calibration scores", xlab="Individual experts", main="")
hist(data$bin, ylab="Calibration scores", xlab="Individual experts", main="")
hist(data$bin2, ylab="Calibration scores", xlab="Individual experts", main="")
hist(data$bin3, ylab="Calibration scores", xlab="Individual experts", main="")
hist(data$bin4, ylab="Calibration scores", xlab="Individual experts", main="")

ggplot() +
  geom_line(data = data, aes(x = expert, y = chi), color = "red") +
  geom_point(data = data, aes(x = expert, y = chi), color = "red") +
  geom_line(data = data, aes(x = expert, y = bin), color = "blue") +
  geom_point(data = data, aes(x = expert, y = bin), color = "blue") +
  geom_line(data = data, aes(x = expert, y = bin2), color = "green") +
  geom_point(data = data, aes(x = expert, y = bin2), color = "green") +
  geom_line(data = data, aes(x = expert, y = bin3), color = "yellow") +
  geom_point(data = data, aes(x = expert, y = bin3), color = "yellow") +
  geom_line(data = data, aes(x = expert, y = bin4), color = "black") +
  geom_point(data = data, aes(x = expert, y = bin4), color = "black") +
  xlab('expert') +
  ylab('calibration score')

colors = c("Chi" = "red", "Bin 1" = "blue", "Bin 2" = "green",
"Bin 3" = "yellow", "Bin 4" = "black")
```

```
ggplot() +
  geom_point(data = gjp_from_than_ten_upto_100_questions, aes(x= 1:2778, y =
  cal_scores_chi_unique_dataACE_GJP, color="Chi")) +
  geom_point(data = gjp_from_than_ten_upto_100_questions, aes(x = 1:2778, y =
  cal_scores_bin_unique_dataACE_GJP, color = "Bin 1")) +
  geom_point(data = gjp_from_than_ten_upto_100_questions, aes(x=1:2778, y =
  cal_scores_bin2_unique_dataACE_GJP, color = "Bin 2")) +
  geom_point(data = gjp_from_than_ten_upto_100_questions, aes(x=1:2778, y =
  cal_scores_bin3_unique_dataACE_GJP ,color = "Bin 3")) +
  geom_point(data = gjp_from_than_ten_upto_100_questions, aes(x=1:2778, y =
  cal_scores_bin4_unique_dataACE_GJP, color = "Bin 4")) +
  labs(x = "Expert",
       y = "Calibration score",
       color = "Cal score") +
  scale_color_manual(values = colors)

ggplot() +
  geom_point(data = gjp_more_than_100_questions, aes(x= 1:518, y =
  cal_scores_chi_unique_dataACE_GJP, color="Chi")) +
  geom_point(data = gjp_more_than_100_questions, aes(x = 1:518, y =
  cal_scores_bin_unique_dataACE_GJP, color = "Bin 1")) +
  geom_point(data = gjp_more_than_100_questions, aes(x=1:518, y =
  cal_scores_bin2_unique_dataACE_GJP, color = "Bin 2")) +
  geom_point(data = gjp_more_than_100_questions, aes(x=1:518, y =
  cal_scores_bin3_unique_dataACE_GJP ,color = "Bin 3")) +
  geom_point(data = gjp_more_than_100_questions, aes(x=1:518, y =
  cal_scores_bin4_unique_dataACE_GJP, color = "Bin 4")) +
  labs(x = "Expert",
       y = "Calibration score",
       color = "Cal score") +
  scale_color_manual(values = colors)




cal_scores_chi_unique_dataACE_IDEA = c()
```

```
for (j in 1:length(unique_experts_dataACE_IDEA))
{
  n = compute_n_i(exp_to_bin(experts_dataACE_IDEA,questions_dataACE_IDEA,
  answers_dataACE_IDEA,j))
  s = compute_s_i(exp_to_bin(experts_dataACE_IDEA,questions_dataACE_IDEA,
  answers_dataACE_IDEA,j), outcomes(experts_dataACE_IDEA,
  realizations_dataACE_IDEA,j))
  #print(n)
  #print(s)
  #print("")
  cal_scores_chi_unique_dataACE_IDEA =
  append(cal_scores_chi_unique_dataACE_IDEA, calibration_chi(s, n))
}

cal_scores_bin_unique_dataACE_IDEA = c()
for (j in 1:length(unique_experts_dataACE_IDEA))
{
  n = compute_n_i(exp_to_bin(experts_dataACE_IDEA,questions_dataACE_IDEA,
  answers_dataACE_IDEA,j))
  s = compute_s_i(exp_to_bin(experts_dataACE_IDEA,questions_dataACE_IDEA,
  answers_dataACE_IDEA,j), outcomes(experts_dataACE_IDEA,
  realizations_dataACE_IDEA,j))
  #print(n)
  #print(s)
  #print("")
  cal_scores_bin_unique_dataACE_IDEA =
  append(cal_scores_bin_unique_dataACE_IDEA, calibration_bin_simulations(s, n))
}

cal_scores_bin2_unique_dataACE_IDEA = c()
for (j in 1:length(unique_experts_dataACE_IDEA))
{
  n = compute_n_i(exp_to_bin(experts_dataACE_IDEA,questions_dataACE_IDEA,
  answers_dataACE_IDEA,j))
  s = compute_s_i(exp_to_bin(experts_dataACE_IDEA,questions_dataACE_IDEA,
  answers_dataACE_IDEA,j), outcomes(experts_dataACE_IDEA,
```

```
  realizations_dataACE_IDEA,j))
  #print(n)
  #print(s)
  #print("")
  cal_scores_bin2_unique_dataACE_IDEA =
  append(cal_scores_bin2_unique_dataACE_IDEA, calibration_bin_simulations2(s, n)
}

cal_scores_bin3_unique_dataACE_IDEA = c()
for (j in 1:length(unique_experts_dataACE_IDEA))
{
  n = compute_n_i(exp_to_bin(experts_dataACE_IDEA,questions_dataACE_IDEA,
  answers_dataACE_IDEA,j))
  s = compute_s_i(exp_to_bin(experts_dataACE_IDEA,questions_dataACE_IDEA,
  answers_dataACE_IDEA,j), outcomes(experts_dataACE_IDEA, realizations_dataACE_ID
  #print(n)
  #print(s)
  #print("")
  cal_scores_bin3_unique_dataACE_IDEA =
  append(cal_scores_bin3_unique_dataACE_IDEA, calibration_bin_simulations3(s, n)
}

cal_scores_bin4_unique_dataACE_IDEA = c()
for (j in 1:length(unique_experts_dataACE_IDEA))
{
  n = compute_n_i(exp_to_bin(experts_dataACE_IDEA,questions_dataACE_IDEA,
  answers_dataACE_IDEA,j))
  s =
compute_s_i(exp_to_bin(experts_dataACE_IDEA,questions_dataACE_IDEA,
  answers_dataACE_IDEA,j),
  outcomes(experts_dataACE_IDEA,
  realizations_dataACE_IDEA,j)),
  cal_scores_bin4_unique_dataACE_IDEA =
  append(cal_scores_bin4_unique_dataACE_IDEA, calibration_bin_simulations4(s, n)
}
```

```
data2 <- data.frame(
  expert = unique_experts_dataACE_IDEA,
  chi = cal_scores_chi_unique_dataACE_IDEA,
  bin = cal_scores_bin_unique_dataACE_IDEA,
  bin2 = cal_scores_bin2_unique_dataACE_IDEA,
  bin3 = cal_scores_bin3_unique_dataACE_IDEA,
  bin4 = cal_scores_bin4_unique_dataACE_IDEA
)

idea_more_than_nine_questions=calibr_scores_dataACE_IDEA_file
[calibr_scores_dataACE_IDEA_file$exp_questions_dataACE_IDEA > 9,]

exp_questions_dataACE_IDEA<-sapply(1:length(unique_experts_dataACE_IDEA),
function(x) length(which(experts_dataACE_IDEA==(unique_experts_dataACE_IDEA[x])))

calibr_scores_dataACE_IDEA<-cbind(unique_experts_dataACE_IDEA,
exp_questions_dataACE_IDEA, cal_scores_chi_unique_dataACE_IDEA,
cal_scores_bin_unique_dataACE_IDEA, cal_scores_bin2_unique_dataACE_IDEA,
cal_scores_bin3_unique_dataACE_IDEA, cal_scores_bin4_unique_dataACE_IDEA)

hist(data2$chi, ylab="Calibration scores", xlab="Individual experts", main="")
hist(data2$bin, ylab="Calibration scores", xlab="Individual experts", main="")
hist(data2$bin2, ylab="Calibration scores", xlab="Individual experts", main="")
hist(data2$bin3, ylab="Calibration scores", xlab="Individual experts", main="")
hist(data2$bin4, ylab="Calibration scores", xlab="Individual experts", main="")

colors2 = c("Chi" = "red", "Bin 1" = "blue", "Bin 2" = "green", "Bin 3" =
"yellow", "Bin 4" = "black")

ggplot() +
  geom_point(data = idea_more_than_nine_questions, aes(x= 1:84, y =
  cal_scores_chi_unique_dataACE_IDEA, color="Chi")) +
  geom_point(data = idea_more_than_nine_questions, aes(x = 1:84, y =
  cal_scores_bin_unique_dataACE_IDEA, color = "Bin 1")) +
  geom_point(data = idea_more_than_nine_questions, aes(x=1:84, y =
  cal_scores_bin2_unique_dataACE_IDEA, color = "Bin 2")) +
```

```
  geom_point(data = idea_more_than_nine_questions, aes(x=1:84, y =
  cal_scores_bin3_unique_dataACE_IDEA ,color = "Bin 3")) +
  geom_point(data = idea_more_than_nine_questions, aes(x=1:84, y =
  cal_scores_bin4_unique_dataACE_IDEA, color = "Bin 4")) +
  labs(x = "Expert",
       y = "Calibration score",
       color = "Cal score") +
  scale_color_manual(values = colors2)


cal_scores_chi_unique_dataSIPS = c()
for (j in 1:length(unique_experts_dataSIPS))
{
  n = compute_n_i(exp_to_bin(experts_dataSIPS,questions_dataSIPS,
  answers_dataSIPS,j))
  s = compute_s_i(exp_to_bin(experts_dataSIPS,questions_dataSIPS,
  answers_dataSIPS,j), outcomes(experts_dataSIPS, realizations_dataSIPS,j))
  cal_scores_chi_unique_dataSIPS = append(cal_scores_chi_unique_dataSIPS,
  calibration_chi(s, n))
}

cal_scores_bin_unique_dataSIPS = c()
for (j in 1:length(unique_experts_dataSIPS))
{
  n = compute_n_i(exp_to_bin(experts_dataSIPS,questions_dataSIPS,
  answers_dataSIPS,j))
  s = compute_s_i(exp_to_bin(experts_dataSIPS,questions_dataSIPS,
  answers_dataSIPS,j), outcomes(experts_dataSIPS, realizations_dataSIPS,j))
  cal_scores_bin_unique_dataSIPS = append(cal_scores_bin_unique_dataSIPS,
  calibration_bin_simulations(s, n))
}

cal_scores_bin2_unique_dataSIPS = c()
for (j in 1:length(unique_experts_dataSIPS))
{
  n = compute_n_i(exp_to_bin(experts_dataSIPS,questions_dataSIPS,
```

```
  answers_dataSIPS,j))
  s = compute_s_i(exp_to_bin(experts_dataSIPS,questions_dataSIPS,
  answers_dataSIPS,j), outcomes(experts_dataSIPS, realizations_dataSIPS,j))
  cal_scores_bin2_unique_dataSIPS = append(cal_scores_bin2_unique_dataSIPS,
  calibration_bin_simulations2(s, n))
}

cal_scores_bin3_unique_dataSIPS = c()
for (j in 1:length(unique_experts_dataSIPS))
{
  n = compute_n_i(exp_to_bin(experts_dataSIPS,questions_dataSIPS,
  answers_dataSIPS,j))
  s = compute_s_i(exp_to_bin(experts_dataSIPS,questions_dataSIPS,
  answers_dataSIPS,j), outcomes(experts_dataSIPS, realizations_dataSIPS,j))
  cal_scores_bin3_unique_dataSIPS = append(cal_scores_bin3_unique_dataSIPS,
  calibration_bin_simulations3(s, n))
}

cal_scores_bin4_unique_dataSIPS = c()
for (j in 1:length(unique_experts_dataSIPS))
{
  n = compute_n_i(exp_to_bin(experts_dataSIPS,questions_dataSIPS,
  answers_dataSIPS,j))
  s = compute_s_i(exp_to_bin(experts_dataSIPS,questions_dataSIPS,
  answers_dataSIPS,j), outcomes(experts_dataSIPS, realizations_dataSIPS,j))
  cal_scores_bin4_unique_dataSIPS = append(cal_scores_bin4_unique_dataSIPS,
  calibration_bin_simulations4(s, n))
}

data3 <- data.frame(
  expert = unique_experts_dataSIPS,
  chi = cal_scores_chi_unique_dataSIPS,
  bin = cal_scores_bin_unique_dataSIPS,
  bin2 = cal_scores_bin2_unique_dataSIPS,
  bin3 = cal_scores_bin3_unique_dataSIPS,
  bin4 = cal_scores_bin4_unique_dataSIPS
```

```
)

exp_questions_dataSIPS<-sapply(1:length(unique_experts_dataSIPS), function(x)
length(which(experts_dataSIPS==(unique_experts_dataSIPS[x]))))

calibr_scores_dataSIPS<-cbind(unique_experts_dataSIPS, exp_questions_dataSIPS,
cal_scores_chi_unique_dataSIPS, cal_scores_bin_unique_dataSIPS,
cal_scores_bin2_unique_dataSIPS, cal_scores_bin3_unique_dataSIPS,
cal_scores_bin4_unique_dataSIPS)


hist(data3$chi, ylab="Calibration scores", xlab="Individual experts", main="")
hist(data3$bin, ylab="Calibration scores", xlab="Individual experts", main="")
hist(data3$bin2, ylab="Calibration scores", xlab="Individual experts", main="")
hist(data3$bin3, ylab="Calibration scores", xlab="Individual experts", main="")
hist(data3$bin4, ylab="Calibration scores", xlab="Individual experts", main="")



ggplot() +
  geom_line(data = data3, aes(x = expert, y = chi), color = "red") +
  geom_point(data = data3, aes(x = expert, y = chi), color = "red") +
  geom_line(data = data3, aes(x = expert, y = bin), color = "blue") +
  geom_point(data = data3, aes(x = expert, y = bin), color = "blue") +
  geom_line(data = data3, aes(x = expert, y = bin2), color = "green") +
  geom_point(data = data3, aes(x = expert, y = bin2), color = "green") +
  geom_line(data = data3, aes(x = expert, y = bin3), color = "yellow") +
  geom_point(data = data3, aes(x = expert, y = bin3), color = "yellow") +
  geom_line(data = data3, aes(x = expert, y = bin4), color = "black") +
  geom_point(data = data3, aes(x = expert, y = bin4), color = "black") +
  xlab('expert') +
  ylab('calibration score')

ggplot() +
  geom_line(data = data3, aes(x = expert, y = chi, color="red")) +
  geom_point(data = data3, aes(x = expert, y = chi, color= "red")) +
```

```
  geom_line(data = data3, aes(x = expert, y = bin, color = "blue")) +
  geom_point(data = data3, aes(x = expert, y = bin, color = "blue")) +
  geom_line(data = data3, aes(x = expert, y = bin2, color = "green")) +
  geom_point(data = data3, aes(x = expert, y = bin2, color = "green")) +
  geom_line(data = data3, aes(x = expert, y = bin3, color = "yellow")) +
  geom_point(data = data3, aes(x = expert, y = bin3, color = "yellow")) +
  geom_line(data = data3, aes(x = expert, y = bin4, color = "black")) +
  geom_point(data = data3, aes(x = expert, y = bin4, color = "black")) +
  xlab('expert') +
  ylab('calibration score')

ggplot() +
  geom_line(data = data3, aes(x = expert, y = chi, color="red"), color="red") +
  geom_point(data = data3, aes(x = expert, y = chi, color= "red"), color="red") +
  geom_line(data = data3, aes(x = expert, y = bin, color = "blue")) +
  geom_point(data = data3, aes(x = expert, y = bin, color = "blue")) +
  geom_line(data = data3, aes(x = expert, y = bin2, color = "green")) +
  geom_point(data = data3, aes(x = expert, y = bin2, color = "green")) +
  geom_line(data = data3, aes(x = expert, y = bin3, color = "yellow")) +
  geom_point(data = data3, aes(x = expert, y = bin3, color = "yellow")) +
  geom_line(data = data3, aes(x = expert, y = bin4, color = "black")) +
  geom_point(data = data3, aes(x = expert, y = bin4, color = "black")) +
  xlab('expert') +
  ylab('calibration score')



colors3 = c("Chi" = "red", "Bin 1" = "blue", "Bin 2" = "green", "Bin 3" =
"yellow", "Bin 4" = "black")

ggplot() +
  geom_point(data = data3, aes(x = expert, y = chi, color="Chi")) +
  geom_point(data = data3, aes(x = expert, y = bin, color = "Bin 1")) +
  geom_point(data = data3, aes(x = expert, y = bin2, color = "Bin 2")) +
  geom_point(data = data3, aes(x = expert, y = bin3 ,color = "Bin 3")) +
  geom_point(data = data3, aes(x = expert, y = bin4, color = "Bin 4")) +
```

```
  labs(x = "Expert id",
       y = "Calibration score",
       color = "Cal score") +
  scale_color_manual(values = colors3)



#ggp3_legend <- get_legend(ggp3)
#grid.newpage()
#grid.draw(ggp3_legend)


cal_scores_chi_unique_IG4years = c()
for (j in 1:length(unique_experts_IG4years))
{
  n = compute_n_i(exp_to_bin(experts_IG4years,questions_IG4years,
  answers_IG4years,j))
  s = compute_s_i(exp_to_bin(experts_IG4years,questions_IG4years,
  answers_IG4years,j), outcomes(experts_IG4years, realizations_IG4years,j))
  #print(n)
  #print(s)
  #print("")
  cal_scores_chi_unique_IG4years = append(cal_scores_chi_unique_IG4years,
  calibration_chi(s, n))
}

cal_scores_bin_unique_IG4years = c()
for (j in 1:length(unique_experts_IG4years))
{
  n = compute_n_i(exp_to_bin(experts_IG4years,questions_IG4years,
  answers_IG4years,j))
  s = compute_s_i(exp_to_bin(experts_IG4years,questions_IG4years,
  answers_IG4years,j), outcomes(experts_IG4years, realizations_IG4years,j))
  #print(n)
  #print(s)
  #print("")
```

```
  cal_scores_bin_unique_IG4years = append(cal_scores_bin_unique_IG4years, calibr
}

cal_scores_bin2_unique_IG4years = c()
for (j in 1:length(unique_experts_IG4years))
{
  n = compute_n_i(exp_to_bin(experts_IG4years,questions_IG4years,
  answers_IG4years,j))
  s = compute_s_i(exp_to_bin(experts_IG4years,questions_IG4years,
  answers_IG4years,j), outcomes(experts_IG4years, realizations_IG4years,j))
  #print(n)
  #print(s)
  #print("")
  cal_scores_bin2_unique_IG4years = append(cal_scores_bin2_unique_IG4years,
  calibration_bin_simulations2(s, n))
}

cal_scores_bin3_unique_IG4years = c()
for (j in 1:length(unique_experts_IG4years))
{
  n = compute_n_i(exp_to_bin(experts_IG4years,questions_IG4years,
  answers_IG4years,j))
  s = compute_s_i(exp_to_bin(experts_IG4years,questions_IG4years,
  answers_IG4years,j), outcomes(experts_IG4years, realizations_IG4years,j))
  #print(n)
  #print(s)
  #print("")
  cal_scores_bin3_unique_IG4years = append(cal_scores_bin3_unique_IG4years,
  calibration_bin_simulations3(s, n))
}

cal_scores_bin4_unique_IG4years = c()
for (j in 1:length(unique_experts_IG4years))
{
  n = compute_n_i(exp_to_bin(experts_IG4years,questions_IG4years,
  answers_IG4years,j))
```

```
  s = compute_s_i(exp_to_bin(experts_IG4years,questions_IG4years,
  answers_IG4years,j), outcomes(experts_IG4years, realizations_IG4years,j))
  #print(n)
  #print(s)
  #print("")
  cal_scores_bin4_unique_IG4years = append(cal_scores_bin4_unique_IG4years,
  calibration_bin_simulations4(s, n))
}

data4 <- data.frame(
  expert = unique_experts_IG4years,
  chi = cal_scores_chi_unique_IG4years,
  bin = cal_scores_bin_unique_IG4years,
  bin2 = cal_scores_bin2_unique_IG4years,
  bin3 = cal_scores_bin3_unique_IG4years,
  bin4 = cal_scores_bin4_unique_IG4years
)

ggplot() +
  geom_line(data = data4, aes(x = expert, y = chi), color = "red") +
  geom_point(data = data4, aes(x = expert, y = chi), color = "red") +
  geom_line(data = data4, aes(x = expert, y = bin), color = "blue") +
  geom_point(data = data4, aes(x = expert, y = bin), color = "blue") +
  geom_line(data = data4, aes(x = expert, y = bin2), color = "green") +
  geom_point(data = data4, aes(x = expert, y = bin2), color = "green") +
  geom_line(data = data4, aes(x = expert, y = bin3), color = "yellow") +
  geom_point(data = data4, aes(x = expert, y = bin3), color = "yellow") +
  geom_line(data = data4, aes(x = expert, y = bin4), color = "black") +
  geom_point(data = data4, aes(x = expert, y = bin4), color = "black") +
  xlab('expert') +
  ylab('calibration score')
```

D R code used in appendices A and B

This code was used for (Hanea and Nane, 2019) and no adjustments were made.

```
### Function to get the exact distribution of the binomial sum of ni, pi
#n is a vector n=(n_1,n_2,...,n_k) and p=(p_1,p_2,...,p_k)
get.exact.binsum = function(n,p)
{
        ### Declare a matrix of working space, initialize all values to zero.
        ### The ith row of Psum is the distribution of the sum of the first
    ### i binomials.
        ### Psum[i,j] =  Prob( X1 + ... + Xi = j-1)
        Psum = matrix(0, nrow=length(n), ncol=sum(n)+1)

        ### Start by getting the distribution of the first binomial
        ### i = 1 implicitly
        for(j in 1:(n[1]+1))
        {
                Psum[1,j] = dbinom(j-1,size=n[1],prob=p[1])
        }

        ### For i=2,...|i|, get the distribution of the ith partial
    ###sum by convolution
        for(i in 2:length(n))
        {
                ### Prob( X1 + ... + X1 > n1 + ... + ni) = 0
                nsofar = sum(n[1:i])

                ### For each value j...
                for(j in 1:(nsofar+1))
                {
                        # Prob(X+Y = j) = sum_k( Prob( X = j-k) * Prob( Y = k))
                        Psum[i,j]       = sum(Psum[i-1,1:j]
* dbinom((j:1)-1,size=n[i],prob=p[i]))
                }
        }
```

```
    ### Return the distribution of |i|th partial sum, which
###is the distribution of the complete sum.
    output.pdf = Psum[length(n),]
    output.cdf = cumsum(output.pdf)
    output.s = 0:sum(n)
    output = data.frame(s = output.s, pdf = output.pdf, cdf = output.cdf)

    return(output)
```