Guided Walkthrough for rational agent framework

July 1, 2023

This document walks through how to apply the rational agent framework to your visualization experiment. We provide general guidance and walk through the forecast visualization example in the paper in more detail.

1 General guide for applying the rational agent framework

1.1 Identifying the decision problem

Before we get started to applying the framework to a visualization experiment, we first need to identify the decision-making problem that forms the task in the experiment. Identify the decision problem involves identifying:

- **states** the description of reality we want help users understand with visualization, e.g. whether the temperature is above or below freezing in our weather forecast example.
- **data generating model** the process that generates the data shown in visualizations, from which the distribution of states can be inferred. In the weather forecast example, the data generating model is drawn uniformly from $\{Normal(5,2), Normal(5,3), Normal(5,4), Normal(5,5)\}$.
- an **action space** the report space that the experiment requires users to use, e.g. whether or not to salt the parking lot.
- **signals** the visualizations you show to users for helping them make decisions.
- a **payoff** or **scoring rule** the payoff function used in experiment to incentivize/score users' behaviors. In general, you can think the scoring rule in following two forms.
  - payoff: $A \times \Theta \to \mathbb{R}$, a function mapping action and the state to real-valued payoff. e.g. how many points you lose if you salt when it doesn't freeze or don't salt but it freezes.

  - proper score: $\Delta(\Theta) \times \Theta \to \mathbb{R}$ a function mapping probability distribution over state and the state to real-valued payoff. The proper score applies the optimal action given the belief (the probability distribution), and is a proper measure of the quality of decision.

  There is a one-to-one mapping between payoff and a proper score. We can always construct a proper score from payoff.

Note that the use of scoring rules may have two purposes:

- to incentivize behavioral agents in an experiment, and
- to evaluate the effect of seeing a visualization.

An experimenter may not use the same scoring rule for incentives and for evaluation. For example, the experimenter may use flat payment to reward behavioral agents. Hence, the pre-experimental analysis has two purposes with different use of scoring rules:

- if it is believed that behavioral agents are incentivized by monetary rewards, the experimenter should pre-experimentally calculate the value of information in real money, and evaluate the incentive. The experimenter might prefer to choose a scoring rule with higher incentives compared to a base payment.
- The experimenter can also use a proper scoring rule solely for the purpose of evaluation. The rational baseline, benchmark and value of information can be calculated pre-experimentally to compare with behavioral performance.

In either cases, the scoring rule is taken as input to our framework. The experimenter can select any payoff function, and construct a proper score.

1.2 Pre-experiment analysis

In this section, we will show how to calculate the three pre-experiment quantities (rational baseline, rational benchmark, and value of information) that the rational agent framework offers. The rational agent's prior belief is before seeing the visualization, the unconditional probability of each state is drawn according to the data generating model. Given a visualization signal, the rational agent will update their belief on data generating models based on the joint distribution between visualizations and data generating models. We provide the pseudo code in Alg. 1 and Alg. 2 to calculate the pre-experiment quantities.

For the rational baseline, you should assume that the rational agent has no access to the any signal, i.e. with only the prior belief. Therefore, when simulating the agent on real experiment data (Alg. 1), the agent won't care about the visualization or any provided information but always take the same action based on the prior belief.

Algorithm 1: Rational baseline

Input: the experimental data D with each row representing one experimental trial, prior belief about data generating model $p(d)$, the action space A, and the scoring rule S

Output: the rational baseline $\mathrm{R}_{\varnothing}$

```
belief ← E_{d∼p(d)}(d);
/* the prior belief about states                              */
action ← arg max_{a∼A} E_{θ∼belief}(S(a, θ));
/* the action made on prior belief                            */
payoff ← 0;
for row ∈ D do
    dist ← actual data generating model used in row;
    payoff ← payoff + E_{θ∼dist}(S(action, θ));
end
R_∅ ← payoff/# of row in D;
/* rational baseline is the rational agent's expected payoff
   in the experiment when only having prior belief            */
```

For rational benchmark, the rational agent has access to the visualization signal. After seeing the visualization signals, the rational agent will update the belief about states based on the joint distribution between visualizations and data generating models. For example, the visualization signals v provides an information about the possibility that this visualization could occur in data generating models d as $p(v|d)$ and the prior belief about the data generating model is $p(d)$. Then the posterior belief about the data generating models could be $p(d|v) = p(v|d) \cdot p(d)/p(v)$. See the detail algorithm in Alg. 2.

Then you can get the value of information in rational agent framework as the difference between rational benchmark and baseline, i.e. the maximum improvement can be made when agent has the right prior knowledge and follows the Bayesian rule to update the belief.

$$\Delta = \mathrm{R}_{\varnothing} - \mathrm{R}_V$$

1.3 Post-experiment analysis

The rational agent framework also offers two post-experiment measures (behavioral score and calibrated behavioral score) to quantify behavioral performance. The behavioral score B is the payoff that the behavioral agents (i.e. real users) get in experiments, which is always below the rational benchmark R_V and can be either above or below the rational baseline $\mathrm{R}_{\varnothing}$. Importantly, if the behavioral score is below the rational baseline, then from the scores alone we cannot reject the hypothesis that the behavioral agent got no useful information from the visualization, because even with no information, the rational agent performs better. Therefore, we propose the calibrated behavioral score, where you can use

Algorithm 2: Rational benchmark

Input: the experimental data D with each row representing one experimental trial, prior belief about data generating model $p(d)$, the action space A, and the scoring rule S

Output: the rational benchmark R_V

$payoff \leftarrow 0$;

for $row \in D$ **do**

- $v \leftarrow$ visualization used in row;
- $p(d|v) = p(v|d) \cdot p(d)/p(v)$;
- `/* the posterior belief about data generating models */`
- $belief \leftarrow E_{d \sim p(d|v)}(d)$;
- `/* the posterior belief about states */`
- $action \leftarrow \arg\max_{a \sim A} E_{\theta \sim belief}(S(a, \theta))$;
- `/* the action made on posterior belief */`
- $dist \leftarrow$ actual data generating model used in row;
- $payoff \leftarrow payoff + E_{\theta \sim dist}(S(action, \theta))$;
- `/* the expected payoff with posterior information */`

end

$\mathrm{R}_V \leftarrow payoff/\#$ of row in D;

the rational agent to calibrated the behavioral score into the interval between the rational baseline and the rational benchmark.

You can follow Alg. 3 to estimate the payoff of behavioral agents' actions.

Algorithm 3: Behavioral score

Input: the experimental data D with each row representing one experimental trial, and the scoring rule S

Output: the behavioral scores *behavioral*

$payoff \leftarrow 0$;

for $row \in D$ **do**

- $dist \leftarrow$ data generating model used in row;
- $action \leftarrow$ action made by the behavioral agent in row;
- $payoff \leftarrow payoff + E_{\theta \sim dist}(S(action, \theta))$;
- `/* the expected score of behavioral action */`

end

$behavioral \leftarrow payoff/\#$ of row in D;

For the calibrated behavioral score, you should simulate the rational agent like calculating the rational baseline and the rational benchmark, but the rational agent is not using the visualizations as signals but using the actions made by behavioral agents, which mean that when seeing a behavioral agent's action a, the rational agent will update the belief about states based on the joint distribution between behavioral actions and data generating models, i.e.

$p(d|a) = p(a|d) \cdot p(d)/p(a)$ (Alg. 4).

Algorithm 4: Calibrated behavioral score

Input: the experimental data D with each row representing one experimental trial, the distribution of data generating models $p(d)$, the action space A, and the scoring rule S

Output: the calibrated behavioral score *benchmark*

$payoff \leftarrow 0$;

for $row \in D$ **do**

  $signal \leftarrow$ action made by the behavioral agent in row

  $p(d|signal) = p(signal|d) \cdot p(d)/p(signal)$;

  `/* the posterior belief of the distribution of data generating models in calibration */`

  $belief \leftarrow E_{d \sim p(d|signal)}(d)$;

  `/* the belief of the data generating models */`

  $action \leftarrow \arg\max_{a \sim A} E_{\theta \sim belief}(S(a, \theta))$;

  `/* the action made on posterior belief */`

  $dist \leftarrow$ data generating model used in row;

  $payoff \leftarrow payoff + E_{\theta \sim dist}(S(action, \theta))$;

  `/* the expected payoff with posterior information */`

end

$benchmark \leftarrow payoff/\#$ of row in D;

2 Walkthrough example: weather forecast visualization

2.1 Setting up for R code

```
> library(dplyr)
> library(tidyr)
> library(modelr)
> knitr::opts_chunk$set(echo = TRUE)
```

2.2 Identifying the decision problem

We will assume an experiment that is a slight variation on the weather forecast example in the paper, where we are asking users to report two actions:

- binary decision: salt the parking lot / not salt
- belief about the possibility of freezing.

The decision-making problem of salting / not salting can be formalized as:

- states: $\theta \in \Theta = \{0 = \text{not freezing}, 1 = \text{freezing}\}$.

- Assume daily low temperature $t \sim N(\mu, \sigma^2)$; $\Pr[\theta = 1] = \Pr[t \leq 0]$; $\mu = 5$ fixed and σ uniformly from $\{2, 3, 4, 5\}$. The data generating models for different visualization strategies:
  - mean: the mean display is not more informative than prior, just knowing $\mu = 5$.
  - CI, gradient, HOPs: seeing the visualization is the same as seeing the distribution of daily low temperature, or equivalently knowing the standard deviation σ of temperature.
- action space: $a \in A = \{0 = \text{not salt}, 1 = \text{salt}\}$
- signals: $v \in V$ showing daily low temperature t.
- scoring rule:
  - payoff: $S_{decision}(a, \theta)$ for reporting the decision of salting, see equation 2 in the paper.
  - proper score: $S_{belief}(b, \theta)$ for reporting the belief of freezing, see eq. 1 below.

$$S_{belief}(b, \theta) = S_{decision}(\arg\max_{a \in A} S_{decision}(a, b), \theta) \tag{1}$$

The following R code defines the scoring rule $S_{decision}$ and S_{belief}. $S_{decision}$ scores the binary action that agents take, where we follow the payoff function in equation 2 in the paper.

```
> belief_space = seq(0, 1, 0.02)
> decision_space = c(0, 1)
> decision_scoring_rule = function(decision, state) {
+   # See equation 3 in the paper
+   -100 * (1 - decision) * state - 10 * decision * (1 - state)
+ }
```

We construct a proper score S_{belief} scoring the belief. S_{belief} is constructed from the payoff $S_{decision}$ by applying the optimal binary decision to the belief, i.e. assuming the agent makes the decision rationally on their belief.

```
> belief_scoring_rule = function(belief, state) {
+   # apply the optimal decision based on the belief
+   decision = decision_space[apply(matrix(unlist(lapply(decision_space, function(d) {
+       decision_scoring_rule(d, belief)
+   })), ncol = length(belief)), 2, which.max)]
+
+   decision_scoring_rule(decision, state)
+ }
```

2.3 For the purposes of this walkthrough only: Simulate behavioral data using quantal response equilibrium

For the purposes of our walkthrough of the rational agent approach, we generate mock experimental data for the weather forecast example described above using quantal response equilibrium[1] (a type of behavioral model that assumes the probability of any particular action being chosen is positively related to the payoff from that action, i.e. very bad actions are unlikely), where we keep the distribution of sigma, i.e. the distribution of data generating models, as uniform distribution.

```
> p_d = c(0.25, 0.25, 0.25, 0.25)
> # p(d) the prior knowledge of the distribution of data generating models
>
> sigma_choices = c(2, 3, 4, 5)
> mu = 5
> data = data.frame(
+   mu = mu,
+   sigma = sigma_choices,
+   vis = c("mean", "mean+interval", "gradient", "HOPs")
+ ) %>%
+   data_grid(mu, sigma, vis) %>%
+   rowwise() %>%
+   mutate(
+     freezing_prob = pnorm(0, mu, sigma),
+     behavioral_decision = list(sample(decision_space, 100,
+       replace = TRUE,
+       # quantal response equilibrium
+       prob = exp(0.2 * decision_scoring_rule(decision_space, freezing_prob)))),
+     behavioral_belief = list(sample(belief_space, 100,
+       replace = TRUE,
+       # quantal response equilibrium
+       prob = exp(belief_scoring_rule(belief_space, freezing_prob))))
+   ) %>%
+   unnest(cols = c(behavioral_decision, behavioral_belief))
> head(data)
```

```
# A tibble: 6 × 6
     mu sigma vis      freezing_prob behavioral_decision behavioral_belief
  <dbl> <dbl> <chr>            <dbl>               <dbl>             <dbl>
1     5     2 gradient       0.00621                   0              0.16
2     5     2 gradient       0.00621                   0              0.36
3     5     2 gradient       0.00621                   0              0.18
4     5     2 gradient       0.00621                   0              0.18
5     5     2 gradient       0.00621                   0              0.32
6     5     2 gradient       0.00621                   0              0.14
```

[1] https://www.sciencedirect.com/science/article/pii/S0899825685710238

2.4 Pre-experiment analysis

For the purpose of walking through the steps entailed in applying the framework, we first show how to calculate the pre-experiment quantities for the weather forecast example by explicit calculations and then walk through it with full code that assumes as behavioral data is the simulated data set generated above.

2.4.1 Explicit Calculations for Weather Forecasting Experiment

Prior We have the prior probability of freezing

$$\begin{aligned} p = \sum\nolimits_{\sigma} \Pr[\theta = 1, \sigma] &= 0.00155 + 0.01195 + 0.0264 + 0.0397 \\ &= 0.0796. \end{aligned}$$

Posterior The posterior probabilities are $\Pr[\theta = 1|\sigma] = 0.62\%, 4.78\%, 10.56\%, 15.87\%$, relatively for $\sigma = 2, 3, 4, 5$. Here we take $\sigma = 2$ as an example:

$$\Pr[\theta = 1|\sigma = 2] = \Pr_{x \sim N(5,2^2)}[x \leq 0] = 0.0062.$$

The rational agent framework gives the following quantities:

rational baseline: $\mathrm{R}_{\varnothing} = -7.96$.

The prior $p = 0.08$ is optimized at no-salt and gives an expected payoff of -7.96. The calculation is as follows:

$$\begin{aligned} \mathrm{R}_{\varnothing} &= \Pr[\theta = 0] \cdot S(a = 0, \theta = 0) + \Pr[\theta = 1] \cdot S(a = 0, \theta = 1) \\ &= (1 - 0.0796) \times 0 + 0.0796 \times (-100) = -7.96 \end{aligned}$$

visualization optimal: $\mathrm{R}_V^{\mathrm{CI}} = \mathrm{R}_V^{\mathrm{gradient}} = \mathrm{R}_V^{\mathrm{HOPs}} = -5.69$; $\mathrm{R}_V^{\mathrm{mean}} = -7.96$.

In CI, gradient, and HOPs, each signal arises with probability 1/4 and the average of the optimal actions under the induced posteriors gives $\mathrm{R}_V = -5.69$. For the visualization of the mean, the rational agent has only the prior information and obtains $\mathrm{R}_V^{\mathrm{mean}} = \mathrm{R}_{\varnothing} = -7.59$.

The calculation of R_V for CI, gradient, and HOPs is the following:

$$\begin{aligned} \mathrm{R}_V^{\mathrm{CI}} = \sum\nolimits_{\sigma,\theta} & \Pr[\sigma, \theta] \cdot S(\mathbf{1}_{\sigma \in \{4,5\}}, \theta) \\ &= 0.24845 \times 0 + 0.00155 \times (-100) + 0.23805 \times 0 \\ &\quad + 0.01195 \times (-100) + 0.2236 \times (-10) + 0.0264 \times 0 \\ &\quad + 0.2103 \times (-10) + 0.0397 \times 0 \\ &= -5.69 \end{aligned}$$

rational benchmark: $\mathrm{R}_I = \max_{\mathrm{vis}} \mathrm{R}_V^{\mathrm{vis}} = -5.69$, the best achievable across visualizations.

value of information: $\Delta = \mathrm{R}_I - \mathrm{R}_{\varnothing} = 2.27$.

2.4.2 Rational baseline

To calculate the rational baseline, we first calculate the prior belief of rational agent and their associated action choice given these beliefs. Then we calculate what score the rational agent would get under the scoring rule if they complete the experiment with only prior knowledge.

```
> prior_belief = Reduce("+",
+   sapply(sigma_choices, function(m) {pnorm(0, mu, m)})) / length(sigma_choices)
> decision_payoffs = lapply(decision_space, function(a) {
+   decision_scoring_rule(a, prior_belief)
+ })
> prior_reporting_decision = decision_space[[which.max(decision_payoffs)]]
> prior_reporting_decision
```

```
[1] 0
```

```
> belief_payoffs = lapply(belief_space, function(b) {
+   belief_scoring_rule(b, prior_belief)
+ })
> prior_reporting_belief = belief_space[[which.max(belief_payoffs)]]
> prior_reporting_belief
```

```
[1] 0
```

```
> # simulate the action of rational agent with only prior knowledge on all experiments
> for (i in 1:nrow(data)) {
+   data[i, "prior_decision_payoff"] =
+     decision_scoring_rule(prior_reporting_decision, data[i, "freezing_prob"])
+   data[i, "prior_belief_payoff"] =
+     belief_scoring_rule(prior_reporting_belief, data[i, "freezing_prob"])
+ }
> prior_payoff = data %>%
+   dplyr::group_by(vis) %>%
+   summarise(prior_decision_payoff = mean(prior_decision_payoff),
+             prior_belief_payoff = mean(prior_belief_payoff))
> prior_payoff
```

```
# A tibble: 4 × 3
  vis           prior_decision_payoff prior_belief_payoff
  <chr>                         <dbl>               <dbl>
1 HOPs                          -7.96               -7.96
2 gradient                      -7.96               -7.96
3 mean                          -7.96               -7.96
4 mean+interval                 -7.96               -7.96
```

The result says the optimal expected payoff of the rational agent with only prior knowledge is -7.96 for both reporting decision and belief, and the optimal decision and belief are both 0, meaning that the rational agent with only prior belief would always report 0 and the rational baseline is -7.96.

2.4.3 Rational benchmark

In the weather forecast example, the visualizations provide perfect information about the variance of data except "mean" visualization, so the updated distribution of data generating models could converge to a situation that only one data generating model has possibility with almost 1 and others are all almost 0. In this case, we assume that the rational agent would know which data generating model is using with certainty after viewing one visualization. Formally, assuming that v provides perfect information about data generating model d_1, then

$$
\begin{gathered}
p(v|d_1) \approx 1 \\
p(v|d_2) \approx 0 \\
\dots \\
p(v|d_n) \approx 0
\end{gathered}
$$

so $p(d_1|v) = p(v|d) \cdot p(d)/p(v) \approx 1$ when the visualization is fixed.

```
> for (i in 1:nrow(data)) {
+   # rational agent can infer the exact data generating model based on signal
+   if (data[[i, "vis"]] == "mean") {
+     posterior_belief = prior_belief
+   } else {
+     posterior_belief = data[[i, "freezing_prob"]]
+   }
+   posterior_reporting_decision = decision_space[[
+     which.max(lapply(decision_space, function(a) {
+       decision_scoring_rule(a, posterior_belief)
+     }))
+   ]]
+   posterior_reporting_belief = belief_space[[
+     which.max(lapply(belief_space, function(b) {
+       belief_scoring_rule(b, posterior_belief)
+     }))
+   ]]
+   data[[i, "posterior_decision_payoff"]] =
+     decision_scoring_rule(posterior_reporting_decision, data[[i, "freezing_prob"]])
+
+   data[[i, "posterior_belief_payoff"]] =
+     belief_scoring_rule(posterior_reporting_belief, data[[i, "freezing_prob"]])
+ }
> posterior_payoff = data %>%
+   dplyr::group_by(vis) %>%
+   summarise(posterior_decision_payoff = mean(posterior_decision_payoff),
+             posterior_belief_payoff = mean(posterior_belief_payoff))
> posterior_payoff
```

```
# A tibble: 4 × 3
  vis           posterior_decision_payoff posterior_belief_payoff
  <chr>                             <dbl>                   <dbl>
1 HOPs                              -5.69                   -5.69
2 gradient                          -5.69                   -5.69
3 mean                              -7.96                   -7.96
4 mean+interval                     -5.69                   -5.69
```

When seeing "mean" visualization, the rational agent didn't get the information about sigma, so they can't update the belief on data generating models, and the rational benchmark for both reporting action and belief turns out to be -7.96, which is the same as rational baseline. For the other visualization, the rational benchmark for both reporting action and belief is -5.69.

2.4.4 Value of information

In the example of weather forecast, the value of information is `posterior_payoff - prior_payoff = 2.27` expect `0` for "mean" visualization.

2.5 Post-experiment analysis

In this section, we will calculate the score of behavioral agents and the score of the calibrated behavioral.

2.5.1 Behavioral score

We use the scoring rule on behavioral agents' actions.

```
> for (i in 1:nrow(data)) {
+   data[i, "behavioral_decision_payoff"] =
+     decision_scoring_rule(data[i, "behavioral_decision"], data[i, "freezing_prob"])
+   data[i, "behavioral_belief_payoff"] =
+     belief_scoring_rule(data[i, "behavioral_belief"], data[i, "freezing_prob"])
+ }
> behavioral_payoff = data %>%
+   dplyr::group_by(vis) %>%
+   summarise(behavioral_decision_payoff = mean(behavioral_decision_payoff),
+             behavioral_belief_payoff = mean(behavioral_belief_payoff))
> behavioral_payoff
```

```
# A tibble: 4 × 3
  vis           behavioral_decision_payoff behavioral_belief_payoff
  <chr>                              <dbl>                    <dbl>
1 HOPs                               -6.92                    -8.60
2 gradient                           -6.81                    -8.57
3 mean                               -6.88                    -8.50
4 mean+interval                      -6.65                    -8.44
```

Since the behavioral agent's actions are generated randomly but not from real persons, some of the payoffs are below rational baseline, e.g. the behavioral payoff when reporting belief. Then we will introduce the calibrated behavioral score, which can represent the behavioral in the baseline-and-benchmark scale.

2.5.2 Calibrated behavioral score

Then we calibrate the behavioral agent.

```
> # the joint distribution of actions and data generating models
> joint_decision_distribution = data %>%
+   dplyr::group_by(mu, sigma, freezing_prob, behavioral_decision) %>%
+   summarise(count = n()) %>%
+   dplyr::group_by(mu, sigma, freezing_prob) %>%
+   mutate(count = count / sum(count))
> joint_belief_distribution = data %>%
+   dplyr::group_by(mu, sigma, freezing_prob, behavioral_belief) %>%
+   summarise(count = n()) %>%
+   dplyr::group_by(mu, sigma, freezing_prob) %>%
+   mutate(count = count / sum(count))
> sum_payoff = 0
> for (i in 1:nrow(data)) {
+   decision_signal = data[[i, "behavioral_decision"]]
+
+   joint_decision_distribution_filtered = joint_decision_distribution %>%
+     filter(behavioral_decision == decision_signal)
+
+   distributions = pnorm(0, mu, sigma_choices)
+
+   p_a = nrow(data %>% filter(behavioral_decision == decision_signal)) / nrow(data)
+
+   sigmas = joint_decision_distribution_filtered$sigma
+   p_a_d = as.list(rep(0, length(sigma_choices)))
+   names(p_a_d) = as.character(sigma_choices)
+   p_a_d[as.character(sigmas)] = joint_decision_distribution_filtered$count  # p(a|d)
+   p_a_d = unname(unlist(p_a_d))
+   p_d_a = p_a_d * p_d / p_a  # p(a|d) * p(d) / p(a)
+
+   calibrated_belief = sum(distributions * p_d_a)  # E_p_d_a(d)
+   calibrated_reporting_decision = decision_space[[which.max(
+     lapply(decision_space, function(a) {
+       decision_scoring_rule(a, calibrated_belief)
+     }))]]
+   data[[i, "calibrated_decision_payoff"]] =
+     decision_scoring_rule(calibrated_reporting_decision, data[[i, "freezing_prob"]])
+
```

```
+
+   belief_signal = data[[i, "behavioral_belief"]]
+
+   joint_belief_distribution_filtered = joint_belief_distribution %>%
+     filter(behavioral_belief == belief_signal)
+
+   p_a = nrow(data %>% filter(behavioral_belief == belief_signal)) / nrow(data)
+
+   sigmas = joint_belief_distribution_filtered$sigma
+   p_a_d = as.list(rep(0, length(sigma_choices)))
+   names(p_a_d) = as.character(sigma_choices)
+   p_a_d[as.character(sigmas)] = joint_belief_distribution_filtered$count  # p(a|d)
+   p_a_d = unname(unlist(p_a_d))
+   p_d_a = p_a_d * p_d / p_a  # p(a|d) * p(d) / p(a)
+
+   calibrated_belief = sum(distributions * p_d_a)  # E_p_d_a(d)
+   calibrated_reporting_belief = belief_space[[which.max(lapply(belief_space, function(a) {
+     belief_scoring_rule(a, calibrated_belief)
+   }))]]
+   data[[i, "calibrated_belief_payoff"]] =
+     belief_scoring_rule(calibrated_reporting_belief, data[[i, "freezing_prob"]])
+ }
> calibrated_payoff = data %>%
+   dplyr::group_by(vis) %>%
+   summarise(calibrated_decision_payoff = mean(calibrated_decision_payoff),
+             calibrated_belief_payoff = mean(calibrated_belief_payoff))
> calibrated_payoff
```

```
# A tibble: 4 × 3
  vis           calibrated_decision_payoff calibrated_belief_payoff
  <chr>                              <dbl>                    <dbl>
1 HOPs                               -6.92                    -5.75
2 gradient                           -6.81                    -5.77
3 mean                               -6.88                    -5.74
4 mean+interval                      -6.65                    -5.77
```

Now we get the calibrated behavioral scores. The calibrated scores of all visualization types are between rational baseline and rational benchmark.