Pilot

Alex Kale

1/15/2019

Visualizations and Heuristics: "Sledgehammer Comparison"

In the document *StimuliAndHeuristics.Rmd*, we list potential visualization formats to test and possible heuristics that users might use to read common language effect size (CLES) from these visualizations. For a pilot experiment, I propose to test a subset of these visualizations: means only, means with intervals, and hypothetical outcome plots (HOPs). Evaluating these visualizations should most directly help us answer the question of whether there is danger in presenting uncertainty in ways that emphasize the mean.

Each of these visualization conditions is associated with possible heuristics.

Means Only: A "Relative Mean Difference" Heuristic

When relying on means alone to make judgments about CLES, users have no uncertainty information. One possible heuristic to judge reliability from means alone is to consider the mean difference relative to the maximum mean difference shown.

$$PerceivedPr(A > B) \propto 50 - 50 * \frac{\mu_B - \mu_A}{\max(|\ \mu_B - \mu_A\ |)}$$

However, this heuristic could also be relative to the axis range.

$$PerceivedPr(A > B) \propto 50 - 50 * \frac{\mu_B - \mu_A}{AxisRange}$$

Means and Intervals: A "Means First, Then Uncertainty" Heuristic or An "Interval Overlap" Heuristic

When we add 95% intervals to the encoding of the group means, we expect that users will rely on this uncertainty information to varying degrees.

If users ignore the intervals completely, we would expect their performance to follow the "relative mean difference" heuristic.

However, based on prior work (Belia 2005, Padilla 2015), we think it is more likely that users will rely on the means as a primary cue and the intervals as a secondary cue, such that the means inform the sense of reliability and interval length informs the baseline for the reliability judgment.

$$PerceivedPr(A > B) \propto 50 - 50 * \frac{\mu_B - \mu_A}{\mu(IntervalLength)/2}$$

Some small number of users may rely on the interval completely, employing a heuristic whereby the overlap between intervals is a cue for the degree to which they should doubt the reliability of the difference between groups. This should be thought of as a piecewise function since the interpretation of the interval overlap depends on which group mean is larger. Where the mean score for team A is larger than the mean score for team B, interval overlap is a cue to the degree to which $A > B$ is uncertain. However, where the mean score for team A is smaller than the mean score for team B, interval overlap is a cue to the degree to which $A > B$ is possible.

$$PerceivedPr(A > B) \propto \begin{cases} 100 - 50 * \frac{IntervalOverlap}{\mu(IntervalLength)} & A \geq B \\ \\ 50 * \frac{IntervalOverlap}{\mu(IntervalLength)} & A < B \end{cases}$$

Alternatively, if axis range is used as a baseline rather than interval length:

$$PerceivedPr(A > B) \propto \begin{cases} 100 - 50 * \frac{IntervalOverlap}{AxisRange} & A \geq B \\ \\ 50 * \frac{IntervalOverlap}{AxisRange} & A < B \end{cases}$$

HOPs: An “Outcome Proportion” Heuristic

With HOPs, the most salient visual cue for reliability is how often the draw for one group is larger than the draw for another. Because HOPs are especially expressive of reliability, this should lead to accurate estimates assuming representative sampling and the sustained attention of the user.

$$PerceivedPr(A > B) \propto 100 * \frac{\Sigma(draws_{A>B})}{\Sigma(draws)}$$

Choosing Data Conditions

To detect reliance on different heuristics, we should evaluate these visualizations using data conditions for which our hypothesized heuristics produce the most disparate estimates. This way the experiment will give us the maximum amount of information about the heurstic a user might be employing. We find these data conditions by searching the space of possible ground truth CLES values (i.e., odds of victory, from which mean differences are derived) and levels of uncertainty (i.e., standard deviations of the group difference distribution) and plotting the heuristic estimates of CLES against the ground truth.

The Space of Possible Data Conditions

We start by setting up the space of possible data conditions in a dataframe.

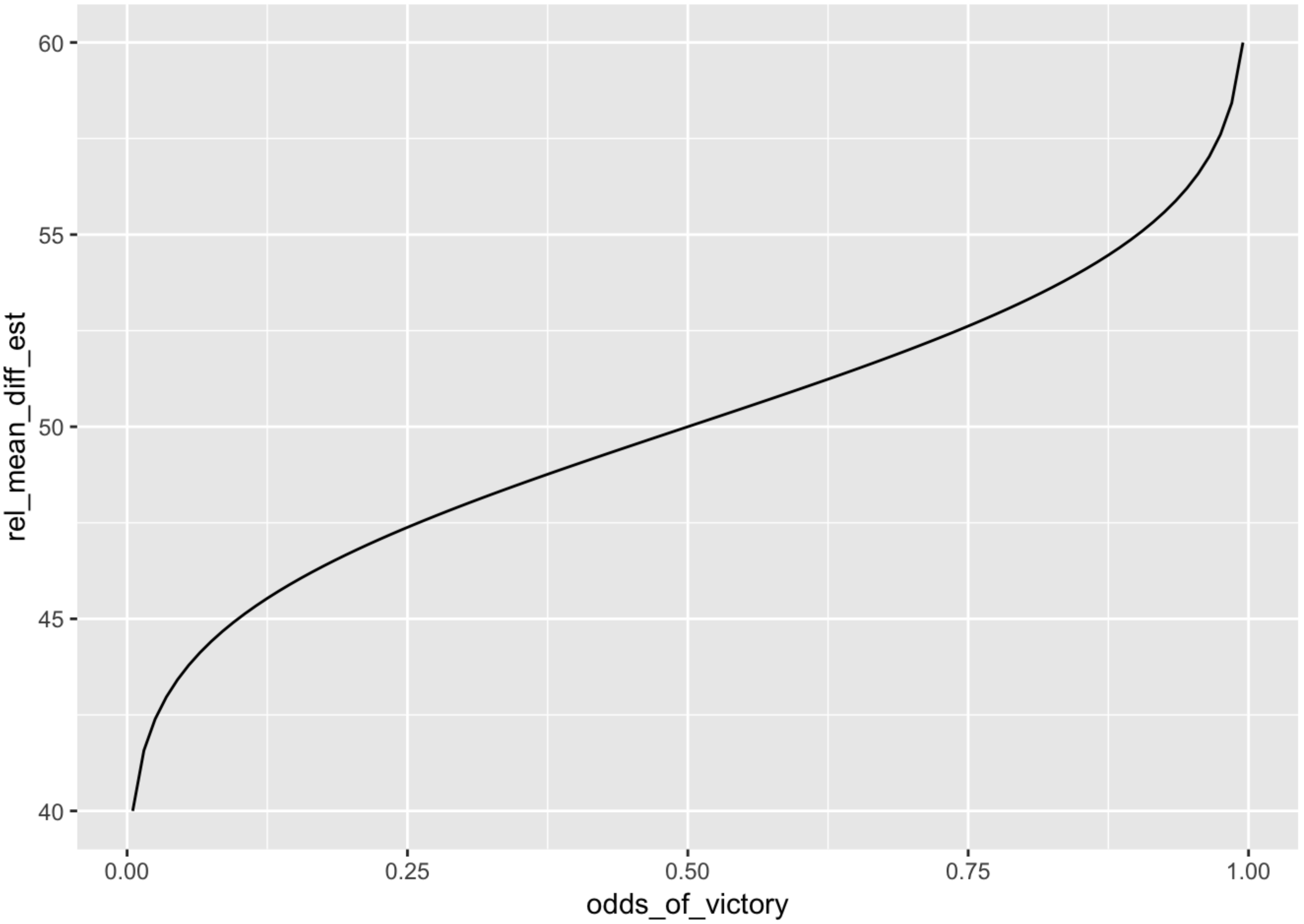
```
# set up possible data conditions dataframe
# std_diff <- seq(0.5, 10, by=0.5) # different levels of uncertainty about the margin of victory
std_diff <- c(1, 5)
odds <- ppoints(100) # probability of team A winning
conds_df <- data.frame(
    "sd_diff" = sort(rep(std_diff, length(odds))),
    "odds_of_victory" = rep(odds, length(std_diff))
)

# add column for the mean difference
conds_df$mean_diff <- - (conds_df$sd_diff * qnorm(conds_df$odds_of_victory)) # mean(B - A)

# add additional columns which are necessary to compute heuristics
conds_df$max_abs_mean_diff <- max(abs(conds_df$mean_diff)) # for the baseline of relative mean difference heuristic
conds_df$sd_team <- sqrt(conds_df$sd_diff ^ 2 / 2) # assume equal and independent variances to get the standard deviation of possible scores for each team for the interval heuristics

# print
head(conds_df)
```

```
##   sd_diff odds_of_victory mean_diff max_abs_mean_diff   sd_team
## 1       1           0.005  2.575829          12.87915 0.7071068
## 2       1           0.015  2.170090          12.87915 0.7071068
## 3       1           0.025  1.959964          12.87915 0.7071068
## 4       1           0.035  1.811911          12.87915 0.7071068
## 5       1           0.045  1.695398          12.87915 0.7071068
## 6       1           0.055  1.598193          12.87915 0.7071068
```

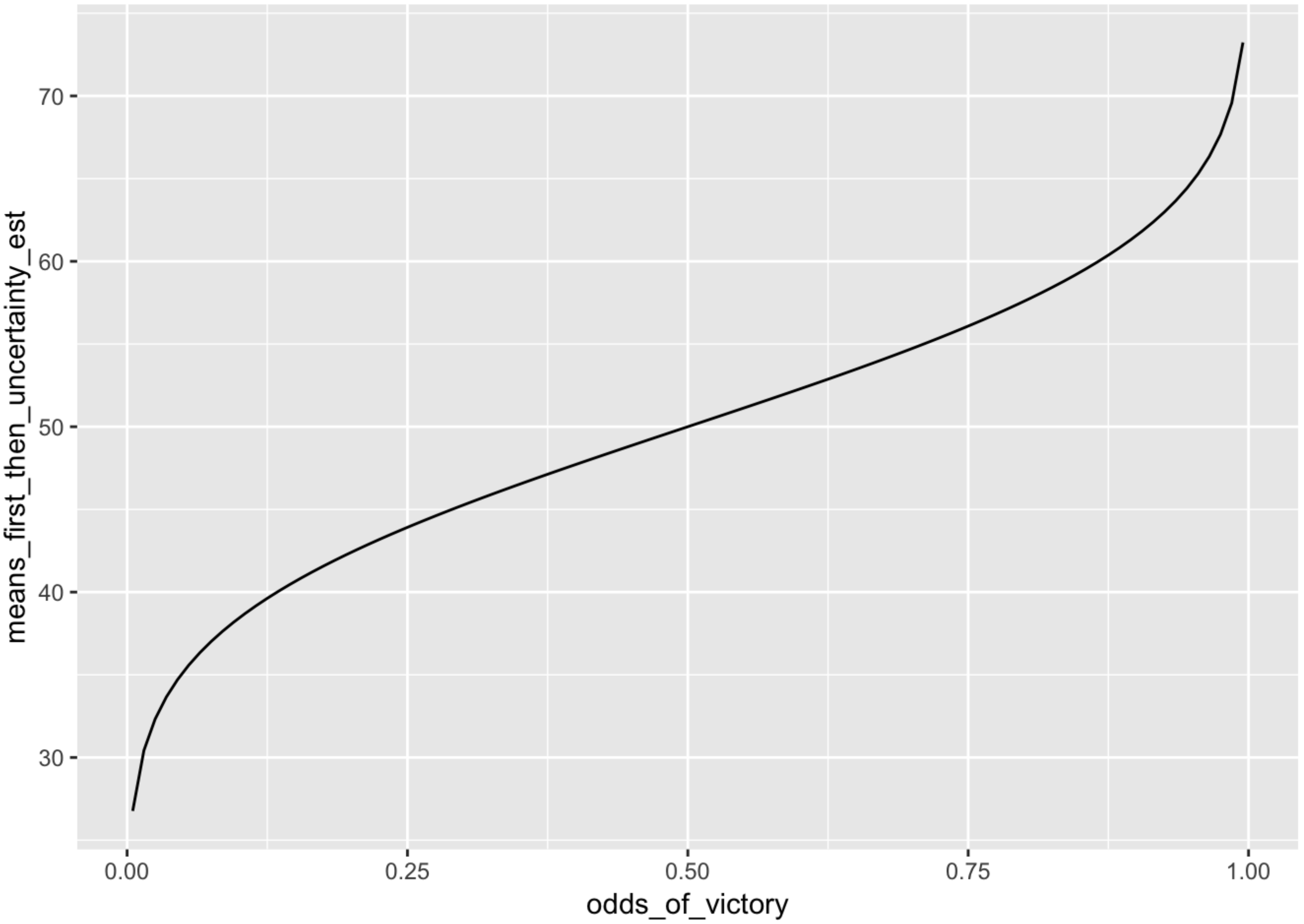
Encoding Heuristics as Functions

Next, we encode the heuristics as functions. For now, we will ignore the possibility that axis range sets the baseline for the reliability judgment.

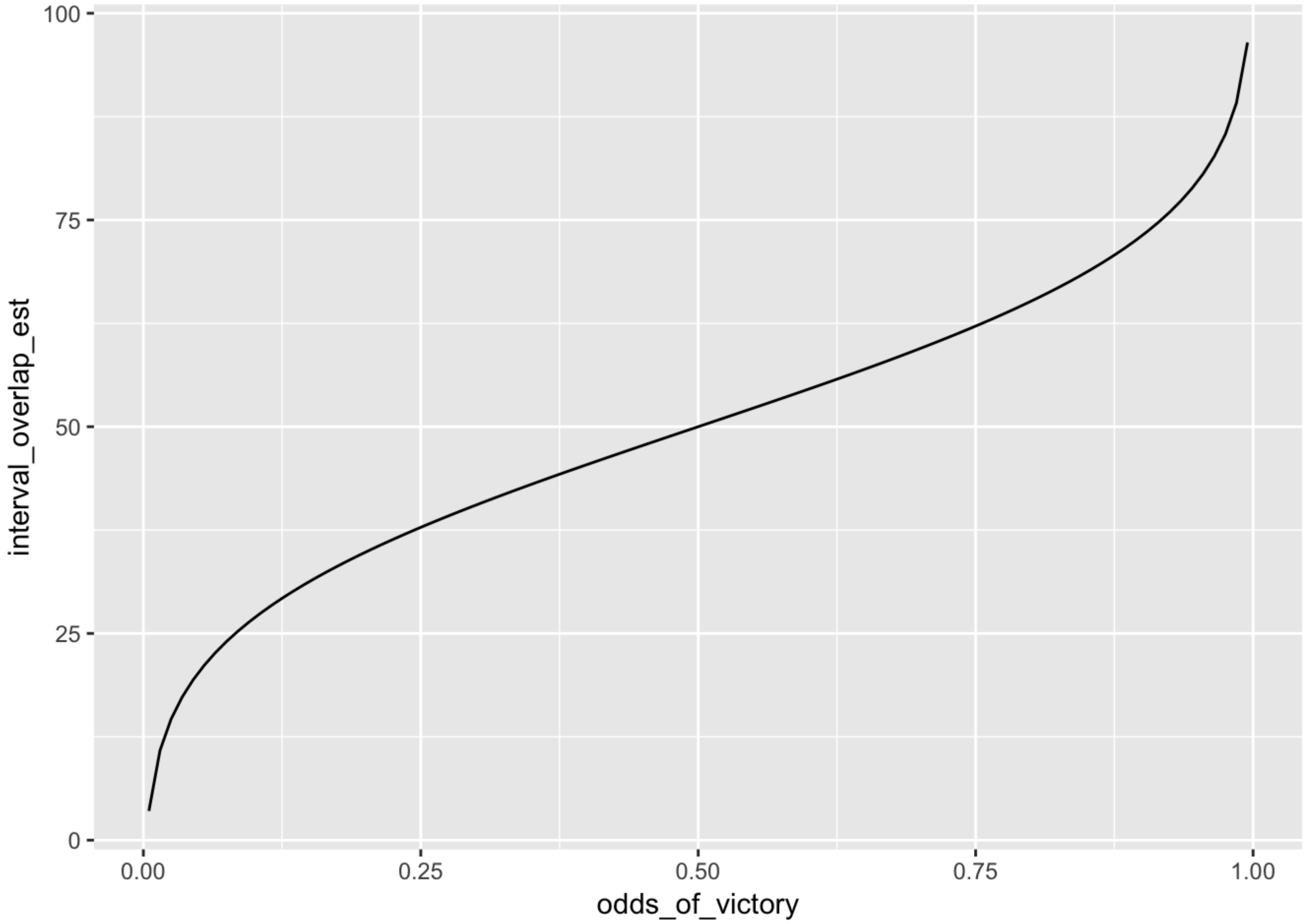
```
# relative mean difference heuristic
relative_mean_difference <- function(mean_diff, max_abs_mean_diff) {
  return(50 - 50 * mean_diff / max_abs_mean_diff)
}
# apply(conds_df, 1, function(df) relative_mean_difference(df['mean_diff'], df['max_abs_mean_diff']))
conds_df %>% filter(sd_diff==1) %>% rowwise() %>% mutate(rel_mean_diff_est = relative_mean_difference(mean_diff, max_abs_mean_diff)) %>% ggplot(aes(x=odds_of_victory,y=rel_mean_diff_est)) + geom_line()
```



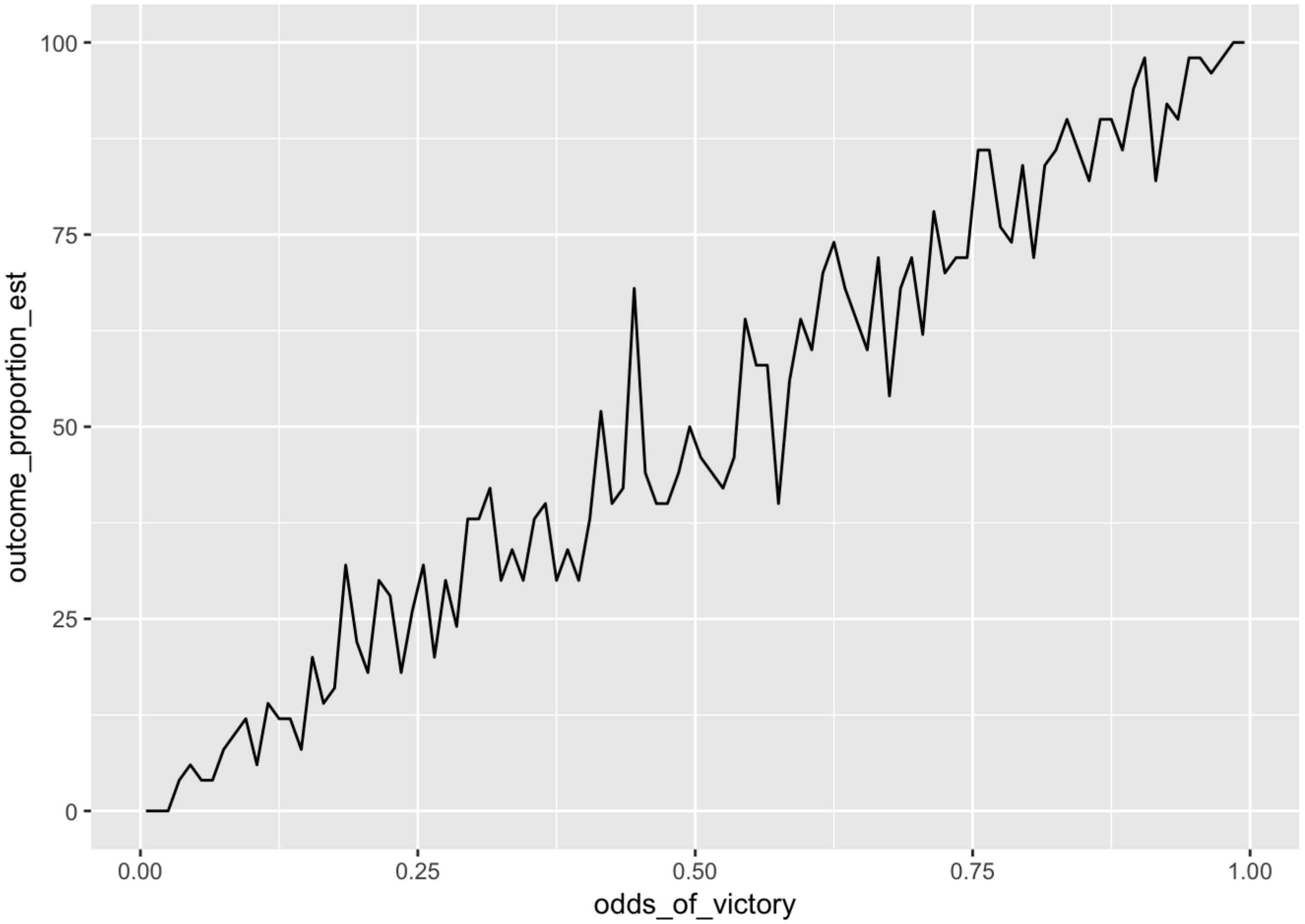
```
# means first, then uncertainty heuristic
means_first_then_uncertainty <- function(mean_diff, sd_team) {
  interval_length <- qnorm(0.975)*sd_team - qnorm(0.025)*sd_team
  return(50 - 50 * mean_diff / interval_length / 2) # assuming that the two intervals are the same length, so we don't need to take their average
}
conds_df %>% filter(sd_diff==1) %>% rowwise() %>% mutate(means_first_then_uncertainty_est = means_first_then_uncertainty(mean_diff, sd_team)) %>% ggplot(aes(x=odds_of_victory,y=means_first_then_uncertainty_est)) + geom_line()
```



```
# interval overlap heuristic
interval_overlap <- function(mean_diff, sd_team) {
  interval_length <- qnorm(0.975)*sd_team - qnorm(0.025)*sd_team # baseline for relative
 judgment (assuming that the two intervals are the same length, so we don't need to take
 their average)
  mean_teamA <- - mean_diff / 2 # relative to center
  mean_teamB <- mean_diff / 2 # relative to center
  # calculation depends on which mean is larger
  if(mean_teamA > mean_teamB) {
    interval_overlap <- (mean_teamB + interval_length / 2) - (mean_teamA - interval_leng
th / 2) # upper bound of lower dist minus lower bound of higher dist
    return(100 - 50 * interval_overlap / interval_length)
  } else { # mean_teamA < mean_teamB
    interval_overlap <- (mean_teamA + interval_length / 2) - (mean_teamB - interval_leng
th / 2) # upper bound of lower dist minus lower bound of higher dist
    return( 50 * interval_overlap / interval_length)
  }
}
conds_df %>% filter(sd_diff==1) %>% rowwise() %>% mutate(interval_overlap_est = interval
_overlap(mean_diff, sd_team)) %>% ggplot(aes(x=odds_of_victory,y=interval_overlap_est))
 + geom_line()
```



```
# outcome proportion heuristic
outcome_proportion <- function(mean_diff, sd_diff) {
  # simulate outcomes (should use the draws participants were actually shown)
  n <- 50
  draws <- rnorm(n, mean_diff, sd_diff)
  return(100 * sum(draws < 0) / n)
}
conds_df %>% filter(sd_diff==1) %>% rowwise() %>% mutate(outcome_proportion_est = outcom
e_proportion(mean_diff, sd_diff)) %>% ggplot(aes(x=odds_of_victory,y=outcome_proportion_
est)) + geom_line()
```

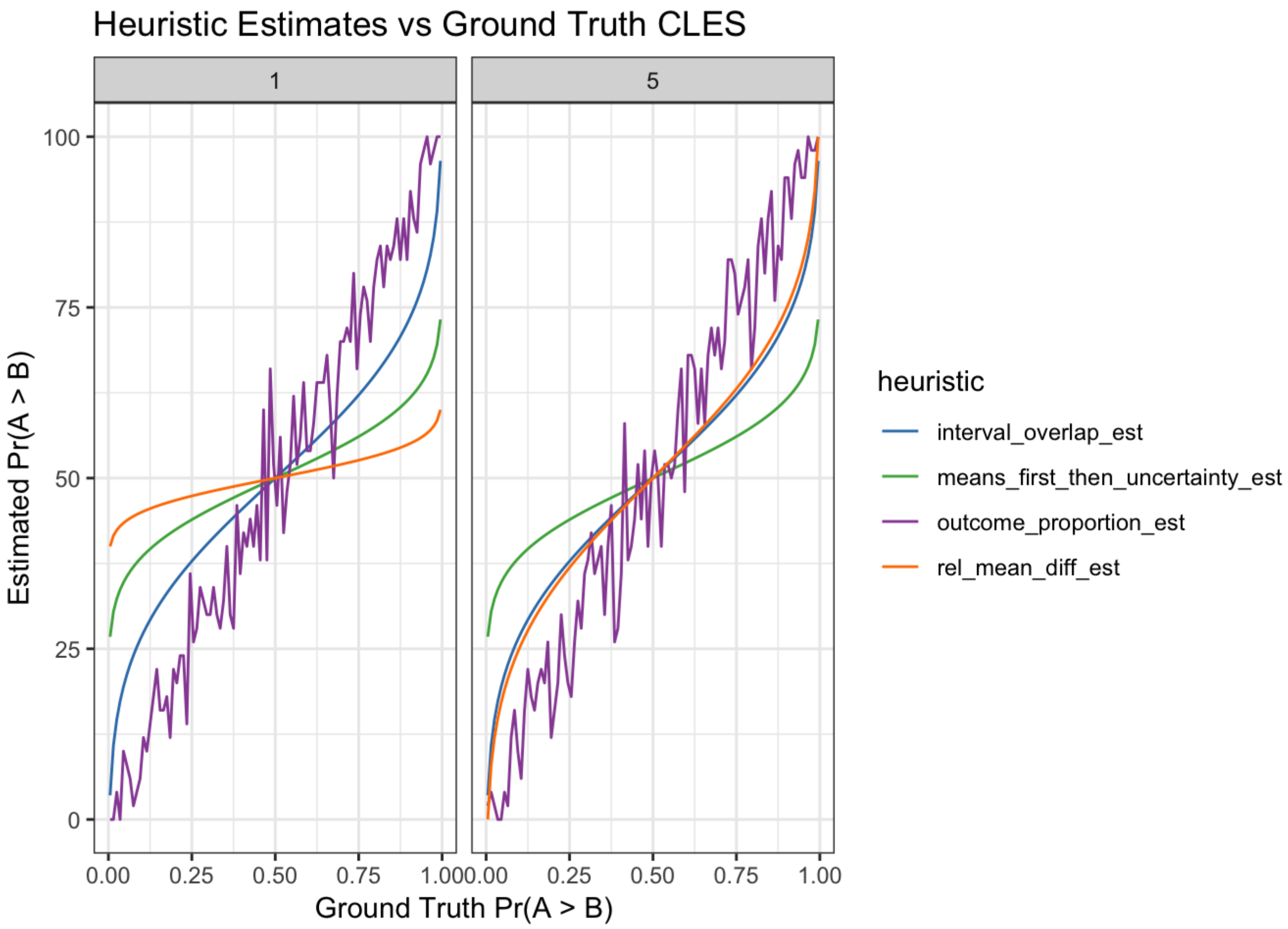


Interestingly, most of these heuristics produce CLES estimates which resemble the π function of prospect theory. I'm not sure what to make of this other than the idea that these visual-spatial heuristics for reading uncertainty information from means and intervals may reinforce or even exacerbate cognitive biases by which people are most sensitive to changes in probability near the extremes of zero and one.

Heuristic Estimates of CLES vs the Ground Truth

Next, we plot the CLES estimates for each of these heuristics against the ground truth $Pr(A > B)$ to see which values of odds of victory $Pr(A > B)$ and standard deviation of the difference σ_{B-A} correspond to the most disparate estimates.

```
conds_df %>% rowwise() %>%
  mutate( # add heuristic estimates
    rel_mean_diff_est = relative_mean_difference(mean_diff, max_abs_mean_diff),
    means_first_then_uncertainty_est = means_first_then_uncertainty(mean_diff, sd_team),
    interval_overlap_est = interval_overlap(mean_diff, sd_team),
    outcome_proportion_est = outcome_proportion(mean_diff, sd_diff)
  ) %>%
  gather(heuristic, est_CLES, rel_mean_diff_est, means_first_then_uncertainty_est, inter
val_overlap_est, outcome_proportion_est) %>% # reshape
ggplot(aes(x = odds_of_victory, y = est_CLES, color = heuristic)) +
  geom_line() +
  colScale +
  theme_bw() +
  labs(title = "Heuristic Estimates vs Ground Truth CLES",
      x = "Ground Truth Pr(A > B)",
      y = "Estimated Pr(A > B)"
  ) +
  facet_wrap(~ sd_diff)
```



One thing to note is that the “relative mean difference” heuristic depends on the maximum mean difference shown which makes it uniquely sensitive to the standard deviation of the difference distribution (B - A), the levels of which are encoded as facets. Notice how only the orange line (i.e., rel_mean_diff_est) is changing across the facets. This tells us that *we don’t need to test very many levels of uncertainty, but we should test a range of ground truth CLES* values (x-axis), especially values near the extremes of the probability scale.

Propagating Heuristic Estimates to Predictions of Betting Behavior

To create a normative model of betting behavior, we simulate an idealized user who, given some estimate for CLES, can perfectly judge the optimal betting amount. This models the impact of distortions in the perception of CLES on betting behavior. Deviations from this normative betting behavior can be captured in bias and noise parameters which account for biases in utility assessment (e.g., risk aversion) and sources of increased error (e.g., task difficulty, the working memory load of reading the vis).

Payoff Scheme and Optimal Betting Behavior

The user bets some portion of their $1 budget in each trial that team A will win the game. The payoff of the bet is proportional to the odds of the bet such that a bet on 1:1 odds yields a 50% chance of winnings double the bet amount, a bet on 2:1 odds yields a 33% chance of winnings tripple the bet amount, a bet on 4:1 odds yields a 20% chance of winnings quintupple the bet amount, etc.

Additionally, the amount that users win is subject to a tiered capital gains tax whereby each increment of 50 cents in winnings is taxed 10% more than the previous 50 cents, and all winnings over $2 are taxed 50%. This tiered tax imposes diminishing returns for excessively risky bets. The amount that users do not bet is subject to a flat tax of 25%, which imposes an incentive against risk aversion much like inflation encourages people to invest in the stock market.

We set this up in the block of code below.

```r
# set range of possible bets based on given budget and minimum bet
budget <- 1
min_bet <- 0.01
possible_bets <- seq(from=min_bet, to=budget, by=0.01)

# create a tiered capital gains tax
tax_winnings <- function(winnings) {
  tiers <- append(seq(0, 2, by = 0.5), Inf)
  rates <- seq(0, .5, by = .1)
  taxed_winnings <- sum(diff(c(0, pmin(winnings, tiers))) * (1-rates))
  return(taxed_winnings)
}

# set cost of not betting
loss_rate <- 0.25
```

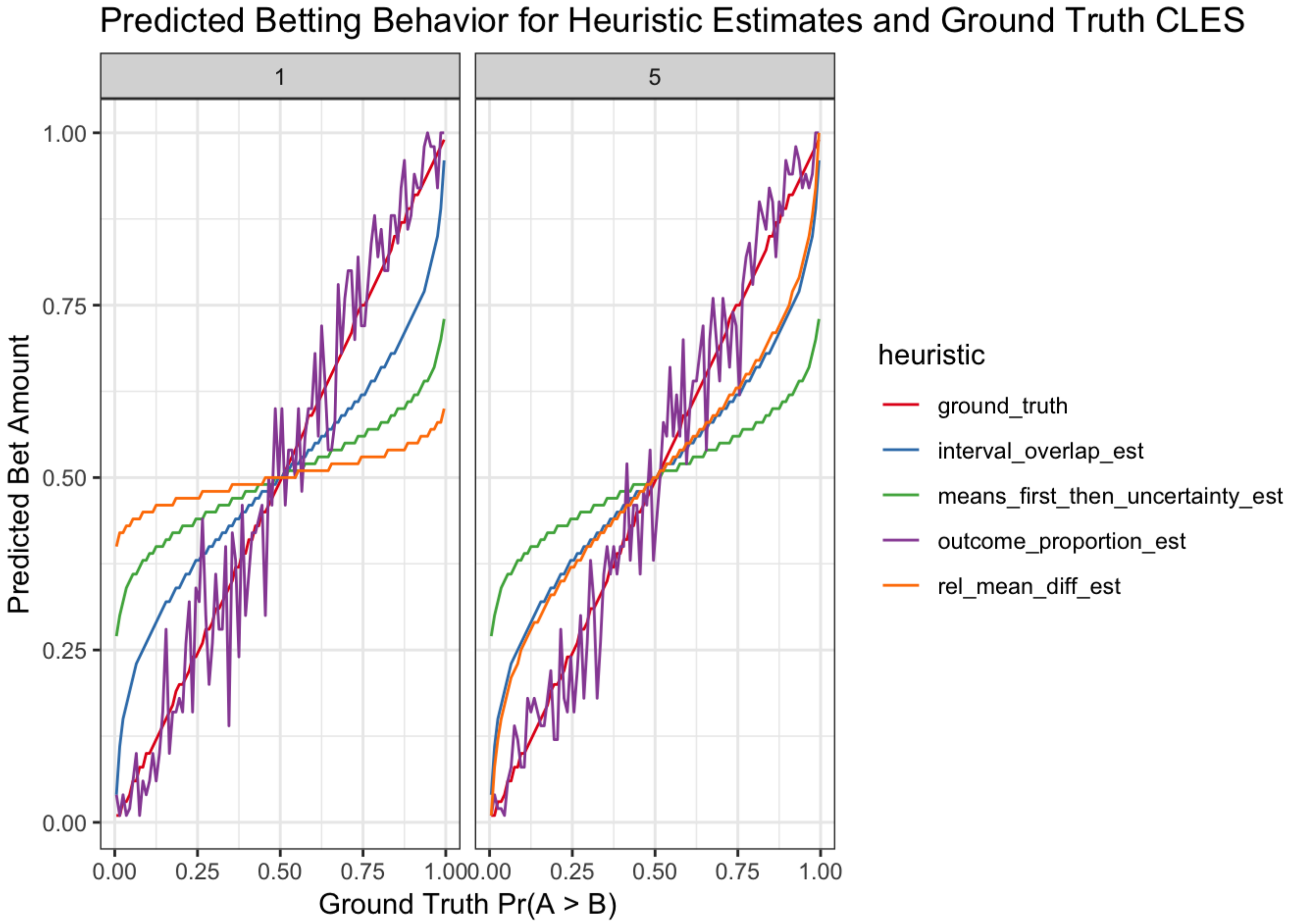
Next, we create a function which determines the optimal bet amount that team A will win for a given $Pr(A > B)$.

```
optimal_bet <- function(p_superiority_A) {
  # hack to catch p == 0
  if (p_superiority_A == 0) {
    p_superiority_A <- 0.001
  }
  # calculate utility over as set of possible bets at the given odds
  utility_in_dollars <- seq(from=-1, to=0, length.out = length(possible_bets))
  for (i in 1:length(possible_bets)) {
    utility_in_dollars[i] <- (1 - loss_rate)*(budget - possible_bets[i]) + p_superiority
_A * tax_winnings(possible_bets[i] / p_superiority_A) # payoff proportional to risk
  }
  # determine the bet with the maximum expected utility
  return(possible_bets[which(utility_in_dollars==max(utility_in_dollars))])
}
```

Predicting Betting Behavior: Toward Building a Model

If we run this function on our heuristic estimates for CLES as well as the ground truth value of CLES, we can see in which data conditions we should expect the most erroneous betting behavior.

```
conds_df %>% rowwise() %>%
  mutate( # add heuristic estimates (as before)
    rel_mean_diff_est = relative_mean_difference(mean_diff, max_abs_mean_diff) / 100, #
 divide by 100 to align scale with ground truth
    means_first_then_uncertainty_est = means_first_then_uncertainty(mean_diff, sd_team)
 / 100,
    interval_overlap_est = interval_overlap(mean_diff, sd_team) / 100,
    outcome_proportion_est = outcome_proportion(mean_diff, sd_diff) / 100,
    ground_truth = odds_of_victory # add ground_truth for purpose of plotting estimated
 bets from ground truth as well as heuristics
  ) %>%
  gather(heuristic, est_CLES, ground_truth, rel_mean_diff_est, means_first_then_uncertai
nty_est, interval_overlap_est, outcome_proportion_est) %>% # reshape, this time includin
g the ground truth as a factor
  rowwise() %>%
  mutate(bet_amount = unlist(map(list(optimal_bet(est_CLES)), 1))) %>% # apply optimal b
et function (if multiple optimal bets, take the lower to avoid error)
ggplot(aes(x = odds_of_victory, y = bet_amount, color = heuristic)) +
  geom_line() +
  colScale +
  theme_bw() +
  labs(title = "Predicted Betting Behavior for Heuristic Estimates and Ground Truth CLE
S",
      x = "Ground Truth Pr(A > B)",
      y = "Predicted Bet Amount"
  ) +
  facet_wrap(~ sd_diff)
```



Note that this is a qualitatively similar pattern to what we see in the CLES estimates. This means that in our task betting behavior (as measured by the amount bet on the outcome $A > B$) should be approximately a linear function of CLES $Pr(A > B)$. Because of the linear relationship between CLES and optimal bet, some of the distortions in betting behavior will follow directly from distortions in the ability to read CLES from the visualization in addition to bias and noise,

$$BetAmount \propto OptimalBet(ReportedCLES) + \beta_{bias}[vis, subject] + \alpha_{noise}[vis, subject]$$

and reported CLES will depend on the heuristic used,

$$ReportedCLES \propto EstCLES_{heuristic}[heuristic] * P_{heuristic}[heuristic, subject]$$

which in turn depends on the visualization condition and the subject.

$$P_{heuristic}[heuristic, subject] \propto \text{logit}(\beta_{vis}[vis] + \alpha_{subject}[subject])$$

The bias parameter in the $BetAmount$ submodel helps us account for individual differences in bias for bet amount (e.g., risk aversion). Additionally, to the extent that we can attribute some of this bias to visualization condition, we might be able to draw interesting conclusions about how different visualization formats influence valuations of utility.

The noise parameter in the $BetAmount$ submodel measures task difficulty, part of which depends on the cognitive load associated with reading CLES from the visualization. To the extent that bet amounts are noisy relative to predicted bets (calculated from $ReportedCLES$) in a manner which is contingent on visualization condition, we can infer that users' working memory is more engaged when reading CLES from some visualizations. This noise should be reduced when users are relying on heuristics (Type 1 processing), and this helps us make the case the heuristics can be a good thing when they don't lead to biased perceptions.

To the extent that visualization condition influences $P_{heuristic}$, we can make the case that users are more likely to use certain heuristics when viewing certain visualizations. This will be essential to our argument about the dangers of emphasizing the mean in uncertainty visualizations.

Copy

Here's how we will frame the task for participants within the experimental interface.

Instructions Page

In this HIT, you will use charts to inform bets on a fictional online game called TeamCrossword.

What is TeamCrossword?

In the popular new online game TeamCrossword, teams of friends work together and attempt to solve crossword puzzles within a time limit. A team scores points for each word they get correct within the time limit. Words are assigned point values depending on their difficulty. In each round of the game, two teams compete against each other, trying to win by earning more points within the time limit.

Betting on TeamCrossword

The creators of TeamCrossword are giving you an opportunity to place bets on game outcomes. *You are assigned to bet on one of two teams each round*, and team names are anonymized to "team A" and "team B" to respect the privacy of TeamCrossword players.

You may bet up to $1 per round of TeamCrossword, and you are payed based on the outcome of the game:

- If the team you bet on wins, you win an amount proportional to the odds that team A will win. For example, 1:1 odds give you a 50% chance of winning 2 times bet amount, 4:1 odds give you a 20% chance of winning 5 times bet amount, 1:3 odds give you a 75% chance to win 1.33 times the bet amount, etc.
- If your team looses the round, you loose the amount that you bet.

Additionally, the game makers introduced some *incentives* to encourage betting and generate additional revenue:

- The amount that you *do not* bet is subject to a flat tax of 25%.
- Winnings in each round are subject to a tiered tax whereby each increment of 50 cents in winnings is taxed 10% more than the previous 50 cents, and all winnings over $2 are taxed 50%.

```
##              tier tax_rate
## 1  0 to 50 cents      10%
## 2 50 cents to $1      20%
## 3      $1 to$1.5      30%
## 4     $1.5 to $2      40%
## 5   more than $2      50%
```

Visualizations of Past Scores

To help you place your bets, the creators of TeamCrossword provide information on how many points each team has scored in previous rounds of TeamCrossword. For each round that you bet on, you are provided a chart of the previous scores of both competing teams (i.e., past scores for team A and team B). *Looking at this information* *allows you to judge the probability of your team winning and improve your bets.*

Practice

Here's an example of one of your charts showing the how many points each competing team has scored in previous rounds of TeamCrossword.

Which team is more likely to win this game?

If this game were to happen 100 times, how many times would team A win?

You have a budget of $1 to bet on the outcome of this game. How much do you bet that team A will win?

Task Page

Task

Trial n out of N

The chart below shows how many points team A and team B have scored in previous rounds of TeamCrossword. Recall that the team with the higher score wins. Use this chart to answer the questions below.

Q1. If this game were to happen 100 times, how many times would you expect team A win?

Q2. You have a budget of $1 to bet on the outcome of this game. How much do you bet that team A will win?

Betting Rules

(This should be displayed somewhere on the task screen as a reminder.)

You may bet up to $1 per round of TeamCrossword:

- If the team you bet on wins, you win an amount proportional to the odds that team A will win.
- If your team looses the round, you loose the amount that you bet.

Incentives:

- The amount that you *do not* bet is subject to a flat tax of 25%.
- Winnings in each round are subject to the tiered tax described in the table below.

```
##             tier tax_rate
## 1  0 to 50 cents      10%
## 2 50 cents to $1      20%
## 3     $1 to $1.5      30%
## 4     $1.5 to $2      40%
## 5   more than $2      50%
```