StimuliGeneration

Alex Kale

8/6/2019

This file contains code to generate stimuli for our effect size judgment and decision-making experiment.

Data Conditions

We manipulate the probability of the team scoring or giving up more points with vs without the new player (p_superiority). We employ two sampling strategies, one which optimizes for each of the two questions we ask participants: 1. Linear intervals in logodds units to give perceptually uniform steps in probability of superiority. 2. Probability of superiority values near the utility optimal decision threshold (i.e., p_superiority == 0.87).

We only sample p_superiority values greater than 0.5, where the new player is expected to *improve* the team's preformance. The decision task is framed as a gain scenario where the user's team needs to score at least 100 points to win an award. In previous pilots, we sample values of p_superiority below 0.5 and framed the decision task as loss aversion. However, we remove these trials to make the task more straightforward.

```
# linear sampling of log odds for full span of ground truth probability of superiority b
etween 0.525 and 0.975
n_trials.full_span <- 10
logodds.full_span <- seq(log(0.525 / (1 - 0.525)), log(0.975 / (1 - 0.975)), length.out
 = n_trials.full_span)

# linear sampling of log odds near the decision threshold (p_superiority == 0.87])
n_trials.near_threshold <- 4
logodds.near_threshold <- seq(log(0.8 / (1 - 0.8)), log(0.9 /(1 - 0.9)), length.out = n_
trials.near_threshold)

# combine the sampling strategies and convert from log odds to probability of superiorit
y
logodds <- sort(c(logodds.full_span, logodds.near_threshold))
p_superiority <- 1 / (1 + exp(-logodds))
n_trials <- length(p_superiority)

print(p_superiority)
```

```
##  [1] 0.5250000 0.6215249 0.7092960 0.7837931 0.8000000 0.8397817 0.8434139
##  [8] 0.8729075 0.8889237 0.9000000 0.9224232 0.9464285 0.9633011 0.9750000
```

We set the baseline probability of winning/keeping the award without the new player to a constant value of 0.5. The team is as likely as a coin flip to win or keep the award without the new player. This represents the scenario where there is the maximum uncertainty about outcomes without intervention.

```
# baseline probability of winning/keeping an award without the new player
baseline <- c(.5) # previously c(.15, .5, 8.5)

# initialize data conditions dataframe
conds_df <- data.frame(
  "p_superiority" = rep(p_superiority, length(baseline)),
  "baseline" = sort(rep(baseline, length(p_superiority))),
  "threshold" = 100)

head(conds_df)
```

```
##   p_superiority baseline threshold
## 1     0.5250000      0.5       100
## 2     0.6215249      0.5       100
## 3     0.7092960      0.5       100
## 4     0.7837931      0.5       100
## 5     0.8000000      0.5       100
## 6     0.8397817      0.5       100
```

We also want to create stimuli for the practice trials. To make these trials easy, we choose probability of superiority values near 1. This way it should be obvious that the new player is worth the cost.

```
# create df containting rows for practice trials
prac_df <- data.frame(
  "p_superiority" = c(.999),
  "baseline" = c(.5),
  "threshold" = c(100))
# append to conditions dataframe
conds_df <- rbind(conds_df, prac_df)

head(prac_df)
```

```
##   p_superiority baseline threshold
## 1         0.999      0.5       100
```

Since judging probability of superiority might be difficult for participants, we are including a mock task to help them understand what we are asking. We ask them judge a case where probability of superiority is 50%.

```
# create df containting rows for mock trial in each condition
mock_df <- data_grid(conds_df, p_superiority = c(.5), baseline = c(.5), threshold = c(10
0))

# add to conds_df
conds_df <- rbind(conds_df, mock_df)

print(mock_df)
```

```
## # A tibble: 1 x 3
##   p_superiority baseline threshold
##           <dbl>    <dbl>     <dbl>
## 1           0.5      0.5       100
```

We control the standard deviation of the distribution of the difference in points between the team with and without the new player (sd_diff) by setting it to 15. In the gain framing this is 15 points scored. In the loss framing, this is 15 points given up. We can think of this variable as constant across trials. We then derive the mean difference in the number of points scored by the team with minus without the new player (mean_diff) from sd_diff and p_superiority.

```
# add columns for the mean and standard deviation of the difference in the number of poi
nts for the team with vs without the new player
conds_df <- conds_df %>%
  mutate(sd_diff = 15, # std(with - without)
         mean_diff = sd_diff * qnorm(p_superiority)) # mean(with - without)

head(conds_df)
```

```
##   p_superiority baseline threshold sd_diff  mean_diff
## 1     0.5250000      0.5       100      15  0.9406017
## 2     0.6215249      0.5       100      15  4.6423208
## 3     0.7092960      0.5       100      15  8.2699398
## 4     0.7837931      0.5       100      15 11.7760197
## 5     0.8000000      0.5       100      15 12.6243185
## 6     0.8397817      0.5       100      15 14.9034139
```

Now we calculate the summary statistics for the team with and without the new player, making the dataframe double its length up to this point. We derive the standard deviation of the points scored by the teams with and without the new player (sd) from sd_diff, variance sum law, and the assumption that the teams with or without the new player have equal and independent variances. We derive the mean number points scored by the teams with and without the new player (mean) from the threshold for winning the award, the sd of points for each version of the team, and the mean_diff between the number of points for with minus without the new player. We derive the probability of winning the award from the threshold, mean, and sd.

```
# double the length of the dataframe to add information per version of the team, with a
 row per distribution to visualize
conds_df <- map_df(seq_len(2), ~conds_df)
conds_df$team <- as.factor(sort(rep(c("With the New Player","Without the New Player"), l
ength(conds_df$p_superiority) / 2)))

# reorder teams for plotting in consistent order
conds_df$Team <- factor(conds_df$team, levels = c("With the New Player","Without the New
Player"))

# add columns for the mean and standard deviation of points for each team and the probab
ility of winning the award
conds_df <- conds_df %>%
  mutate(
    sd = sqrt(conds_df$sd_diff ^ 2 / 2), # assume equal and independent variances
    mean = if_else(Team == "Without the New Player",
                   # team without the new player is at baseline
                   threshold - sd * qnorm(1 - baseline),
                   # team with new player is at difference from baseline
                   threshold - sd * qnorm(1 - baseline) + mean_diff),
    p_award = pnorm((mean - threshold) / sd) # probability of exceeding threshold to win
award
  )

head(conds_df)
```

```
##   p_superiority baseline threshold sd_diff  mean_diff                team
## 1     0.5250000      0.5       100      15  0.9406017 With the New Player
## 2     0.6215249      0.5       100      15  4.6423208 With the New Player
## 3     0.7092960      0.5       100      15  8.2699398 With the New Player
## 4     0.7837931      0.5       100      15 11.7760197 With the New Player
## 5     0.8000000      0.5       100      15 12.6243185 With the New Player
## 6     0.8397817      0.5       100      15 14.9034139 With the New Player
##                  Team      sd     mean   p_award
## 1 With the New Player 10.6066 100.9406 0.5353322
## 2 With the New Player 10.6066 104.6423 0.6691917
## 3 With the New Player 10.6066 108.2699 0.7822155
## 4 With the New Player 10.6066 111.7760 0.8665552
## 5 With the New Player 10.6066 112.6243 0.8830224
## 6 With the New Player 10.6066 114.9034 0.9200053
```

We name the conditions based on the the baseline and probability of superiority, so we can later filter the rows belonging to the same stimulus.

```
# name conditions
conds_df <- conds_df %>%
  rowwise() %>% # need to name each row differently
  mutate(condition = paste(c(baseline, "base", round(p_superiority, 3), "p_sup"), collap
se = "_")) %>%
  ungroup() # need to undo rowwise

head(conds_df)
```

```
## # A tibble: 6 x 11
##   p_superiority baseline threshold sd_diff mean_diff team  Team     sd
##           <dbl>    <dbl>     <dbl>   <dbl>     <dbl> <fct> <fct> <dbl>
## 1         0.525      0.5       100      15     0.941 With… With…  10.6
## 2         0.622      0.5       100      15     4.64  With… With…  10.6
## 3         0.709      0.5       100      15     8.27  With… With…  10.6
## 4         0.784      0.5       100      15    11.8   With… With…  10.6
## 5         0.8        0.5       100      15    12.6   With… With…  10.6
## 6         0.840      0.5       100      15    14.9   With… With…  10.6
## # … with 3 more variables: mean <dbl>, p_award <dbl>, condition <chr>
```

We need to save this dataframe for analysis.

```
# save conds_df with the draws used to create these stimuli (for use in analysis)
save(conds_df, file = "stimuli/conds_df.Rda")
```

Visualization Stimuli

Here, we define functions for each chart type we plan to show users, and we show the practice trial as an example.

First, let's isolate the data we want to plot.

```
# get the data for the gain framing practice trial to use as an example
prac_df <- conds_df %>% filter(p_superiority == 0.999)

head(prac_df)
```

```
## # A tibble: 2 x 11
##   p_superiority baseline threshold sd_diff mean_diff team  Team     sd
##           <dbl>    <dbl>     <dbl>   <dbl>     <dbl> <fct> <fct> <dbl>
## 1         0.999      0.5       100      15      46.4 With… With…  10.6
## 2         0.999      0.5       100      15      46.4 With… With…  10.6
## # … with 3 more variables: mean <dbl>, p_award <dbl>, condition <chr>
```

Before we start building charting functions, we want a helper function to wrap captions and prevent them from running of the edge of our charts.

We also set up some parameters that will remain consistent across charts, including a set x-axis domain, parameters specific to HOPs (i.e., frame rate and number of frames), and sizes for geometries and text, respectively.

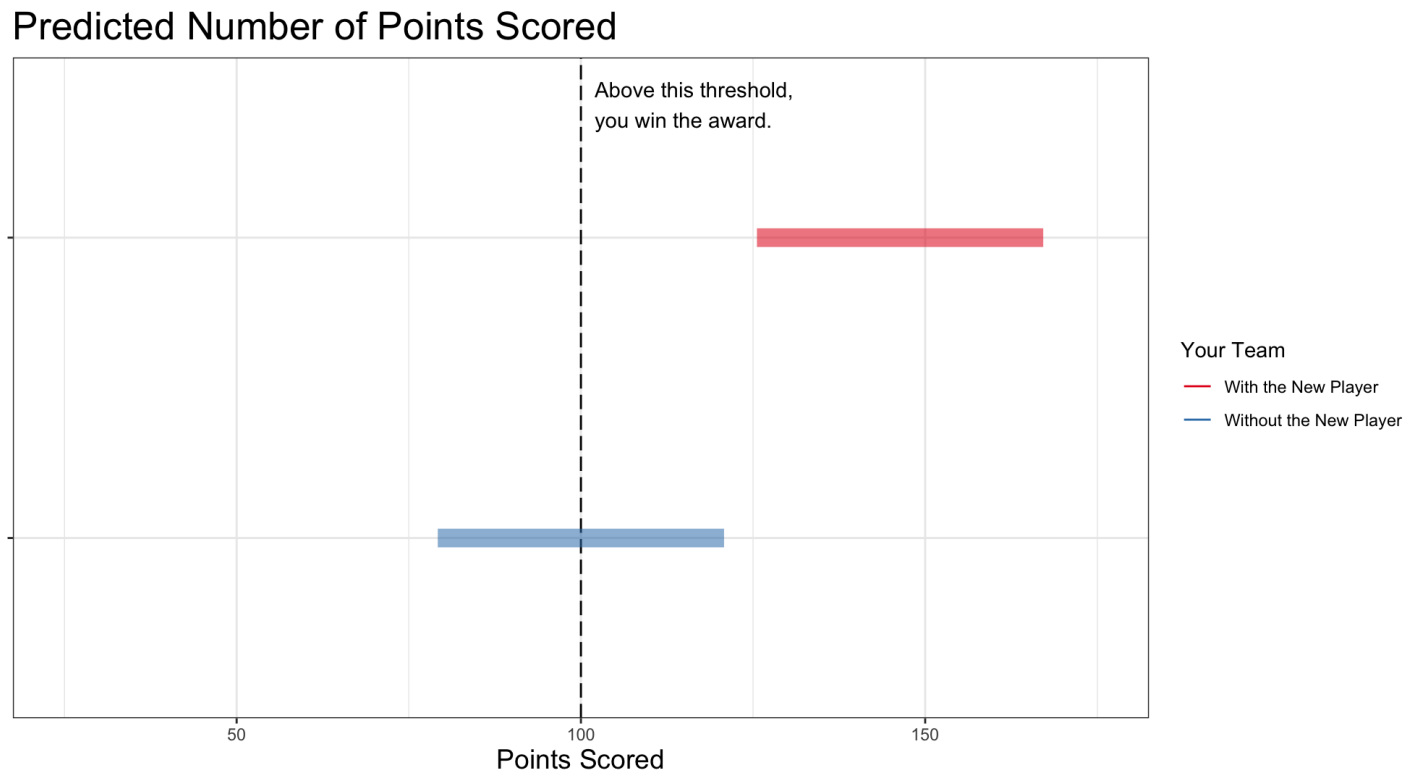
```
# select limits for x-axis
data_domain <- c(25, 175)

# HOPs frame rate
frame_rate <- 2.5
# select number of draws for HOPs conditions
n_draws_hops <- 50
# select number of dots for dotplots
n_dots_qdps <- 20

# geom sizes
means_size_interval <- 12
interval_size <- 2
HOPs_size <- 9
means_size_HOPs <- 12
means_size_slab <- 36

# opacity for uncertainty encodings
opacity <- 0.65

# text formating
title_size <- 20
label_size <- 14
caption_size <- 16
char_before_wrap <- 90

# set plot dimensions
dims_pix <- c(770, 462) # pixel dimensions
ppi <- 75 # assume 75 ppi for the avg monitor
dims <- dims_pix / ppi # dimensions in inches
```

Now we'll create each uncertainty visualization (i.e., intervals, HOPs, quantile dotplots, and densities) with an without extrinsic encodings of the mean.

Intervals

A chart function for visualizations showing only 95% containment intervals.

```
intervals <- function(df, data_domain, title, x_label, caption, decision_threshold, thre
shold_label) {
  plt <- df %>% ggplot(aes(y = mean, x = rev(levels(Team)), color = Team)) +
      geom_hline(yintercept = decision_threshold, linetype = "longdash") + # award thres
hold
      annotate("text", y = (decision_threshold + 2), x = "Without the New Player", label
= threshold_label, hjust = 0, vjust = -2.4) +
      geom_errorbar(aes(ymin = mean + qnorm(0.025) * sd, ymax = mean + qnorm(0.975) * s
d, width = 0, size = interval_size, alpha = opacity)) +
      coord_flip() +
      theme_bw() +
      scale_color_brewer(palette = "Set1") +
      ylim(data_domain[1], data_domain[2]) +
      labs(
        title = title,
        x = NULL,
        y = x_label,
        color = "Your Team",
        caption = wrap_label(caption, char_before_wrap)) +
      guides(
        size = FALSE,
        alpha = FALSE) +
      theme(
        axis.title = element_text(size=label_size),
        axis.text.y = element_blank(),
        plot.title = element_text(size = title_size),
        plot.caption = element_text(size = caption_size, hjust = 0, vjust = -1))

  return(plt)
}

intervals(df = prac_df,
          data_domain = data_domain,
          title = "Predicted Number of Points Scored",
          x_label = "Points Scored",
          caption = "Intervals contain 95% of the possible numbers of points that could
 be scored by your team with (top) and without the new player (bottom).",
          decision_threshold = 100,
          threshold_label = "Above this threshold,\nyou win the award.")
```



Intervals contain 95% of the possible numbers of points that could be scored by your team with (top) and without the new player (bottom).

Intervals With Means

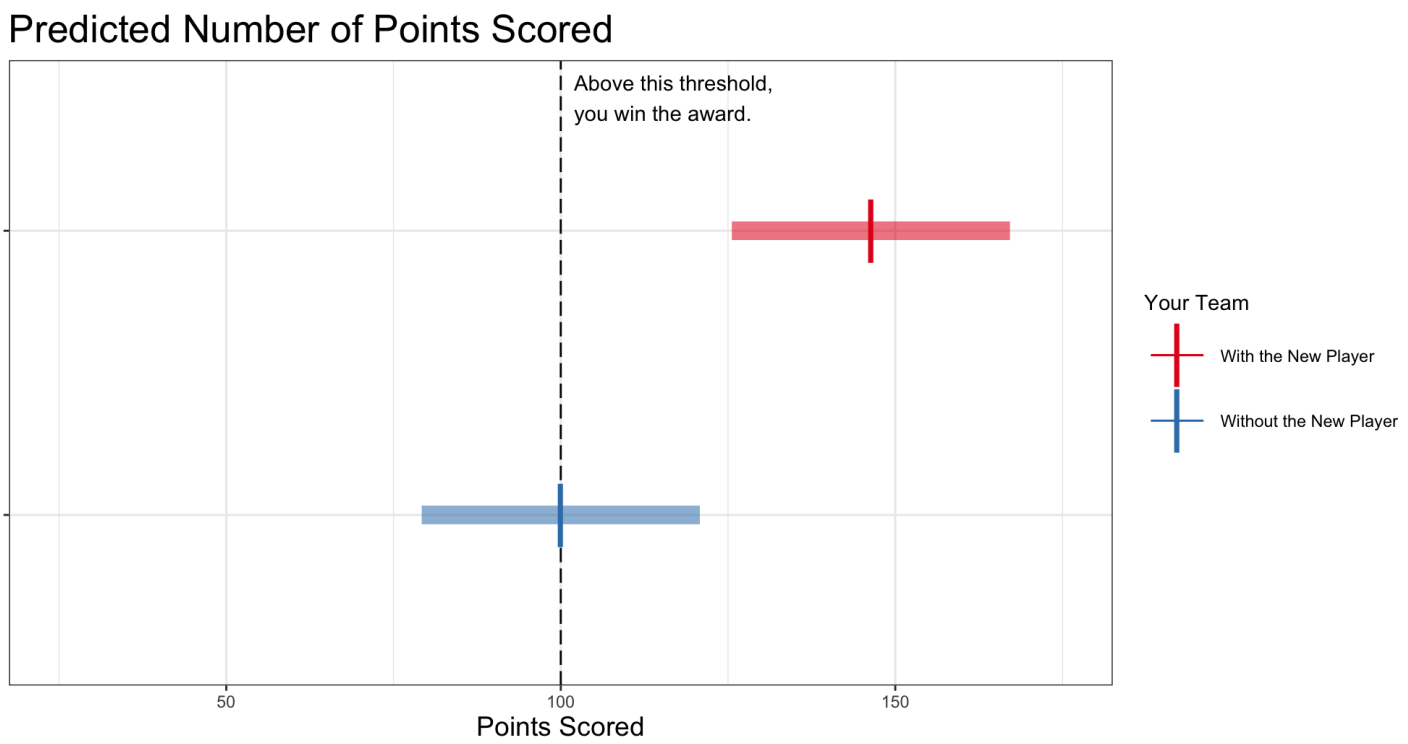
A chart function for visualizations showing only 95% containment intervals with means.

```
intervals_w_means <- function(df, data_domain, title, x_label, caption, decision_thresho
ld, threshold_label, label_mean = FALSE) {
  plt <- df %>% ggplot(aes(y = mean, x = rev(levels(Team)), color = Team)) +
      geom_hline(yintercept = decision_threshold, linetype = "longdash") + # award thres
hold
      annotate("text", y = (decision_threshold + 2), x = "Without the New Player", label
= threshold_label, hjust = 0, vjust = -2.4) +
      geom_errorbar(aes(ymin = mean + qnorm(0.025) * sd, ymax = mean + qnorm(0.975) * s
d, width = 0, size = interval_size, alpha = opacity)) +
      geom_point(aes(y = mean), shape = 124, size = means_size_interval) + # add means
      geom_line(aes(y = mean - 1000)) + geom_point(aes(y = mean - 1000), shape = 124, si
ze = (means_size_interval - 7)) + # hack to get legend symbols oriented properly
      coord_flip() +
      theme_bw() +
      scale_color_brewer(palette = "Set1") +
      ylim(data_domain[1], data_domain[2]) +
      labs(
        title = title,
        x = NULL,
        y = x_label,
        color = "Your Team",
        caption = wrap_label(caption, char_before_wrap)) +
      guides(
        size = FALSE,
        alpha = FALSE
      ) +
      theme(
        axis.title = element_text(size=label_size),
        axis.text.y = element_blank(),
        plot.title = element_text(size = title_size),
        plot.caption = element_text(size = caption_size, hjust = 0, vjust = -1))

  if (label_mean) {
    # get mean positions for each dists
    mean_with <- df %>% filter(Team == "With the New Player") %>% select(mean) %>% as.nu
meric()
    mean_without <- df %>% filter(Team == "Without the New Player") %>% select(mean) %>%
as.numeric()

    plt <- plt +
      geom_segment(x = 1.9, xend = 1.1, y = mean_with, yend = 146.3535, size = 0.5, line
type = "longdash", color = "black") +
      geom_segment(x = 1.1, xend = 1.1, y = mean_without, yend = 146.3535, size = 0.5, l
inetype = "longdash", color = "black") +
      annotate("text", x = 1.1, y = 148, label = "Average\npredicted\nscore", hjust = 0,
vjust = 0.5)
  }

  return(plt)
}

intervals_w_means(df = prac_df,
                  data_domain = data_domain,
```

```
                  title = "Predicted Number of Points Scored",
                  x_label = "Points Scored",
                  caption = "The red and blue vertical lines on each plot represent the
 average number of points that could be scored by your team with (top) and without the n
ew player (bottom). Intervals contain 95% of the possible numbers of points that could b
e scored.",
                  decision_threshold = 100,
                  threshold_label = "Above this threshold,\nyou win the award.")
```

```
## Warning: Removed 2 rows containing missing values (geom_path).
```

```
## Warning: Removed 2 rows containing missing values (geom_point).
```



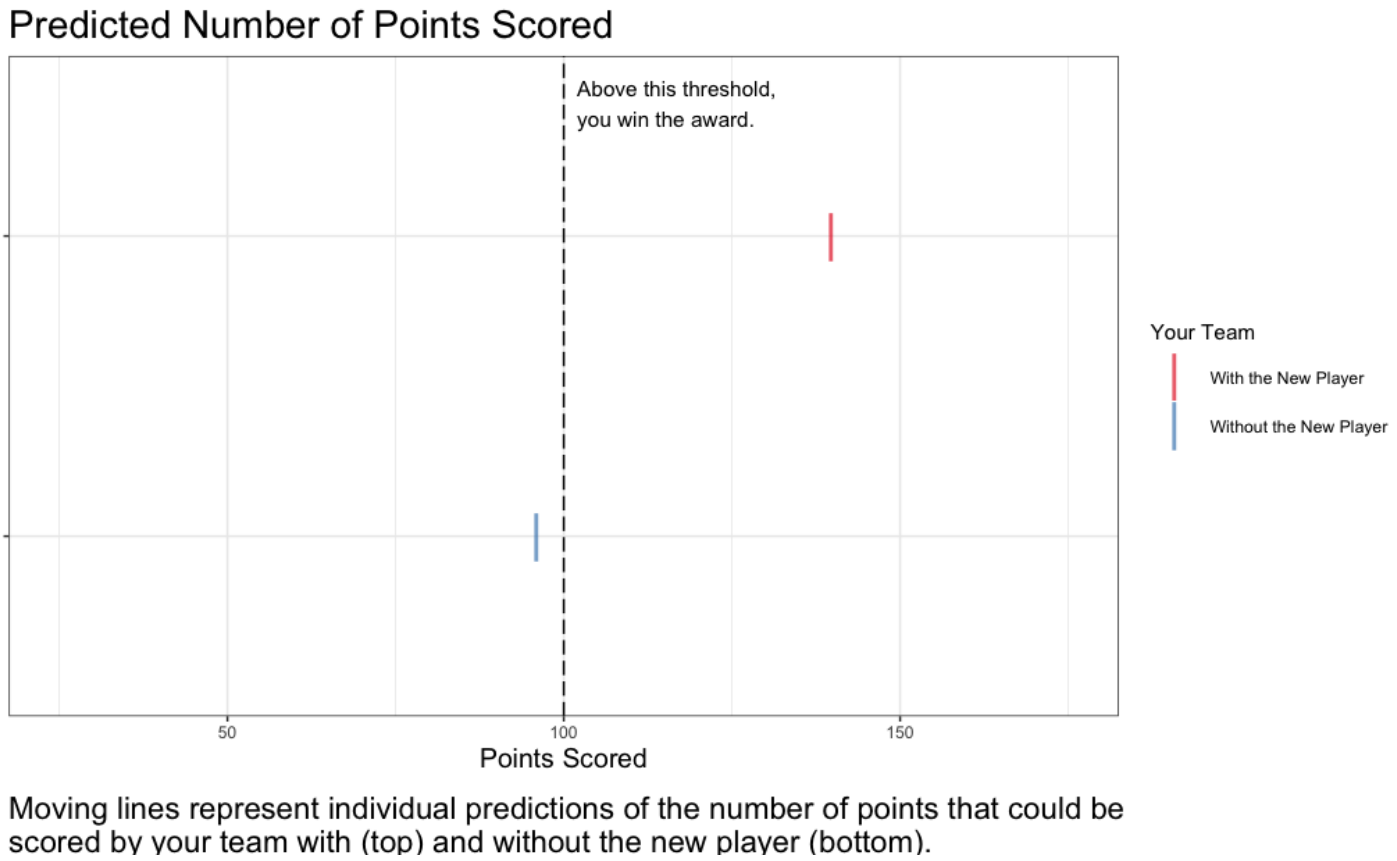
The red and blue vertical lines on each plot represent the average number of points that could be scored by your team with (top) and without the new player (bottom). Intervals contain 95% of the possible numbers of points that could be scored.

Hypothetical Outcome Plots (HOPs)

A chart function for HOPs of the possible points for each version of the team.

```
hops <- function(df, n_draws, frames_per_second, data_domain, title, x_label, caption, d
ecision_threshold, threshold_label, dimensions) {
  plt <- df %>%
      mutate(
        quantiles = list(sample(ppoints(n_draws))), # get n_draws equally spaced quantil
es and shuffle their order
        draws = pmap(list(quantiles, mean, sd), qnorm), # map those quantiles to draws f
rom each distribution
        draw_n = if_else(Team == "With the New Player", # number draws differently for e
ach distribution so we aren't showing the same quantile for both distributions
                         list(sample(1:n_draws)),
                         list(1:n_draws))
      ) %>%
      unnest(cols = c(quantiles, draws, draw_n)) %>% # one row per draw
      ggplot(aes(y = draws, x = Team, color = Team)) +
      scale_x_discrete(limits = rev(levels(df$Team))) + # get teams in correct order (re
d on top)
      geom_hline(yintercept = decision_threshold, linetype = "longdash") + # award thres
hold
      annotate("text", y = (decision_threshold + 2), x = "With the New Player", label =
 threshold_label, hjust = 0, vjust = -2.4) +
      geom_point(shape = 124, size = HOPs_size, alpha = opacity) +
      coord_flip() +
      theme_bw() +
      scale_color_brewer(palette = "Set1") +
      ylim(data_domain[1], data_domain[2]) +
      labs(
        title = title,
        x = NULL,
        y = x_label,
        color = "Your Team",
        caption = wrap_label(caption, char_before_wrap)) +
      theme(
        axis.title = element_text(size=label_size),
        axis.text.y = element_blank(),
        plot.title = element_text(size = title_size),
        plot.caption = element_text(size = caption_size, hjust = 0, vjust = -1)) +
      transition_manual(draw_n)

  animation <- animate(plt, fps = frames_per_second, nframes = 10 * frames_per_second, r
es = 100, width = dimensions[1]*100, height = dimensions[2]*100)

  return(animation)
}

hops(df = prac_df,
     n_draws = n_draws_hops,
     frames_per_second = frame_rate,
     data_domain = data_domain,
     title = "Predicted Number of Points Scored",
     x_label = "Points Scored",
     caption = "Moving lines represent individual predictions of the number of points th
at could be scored by your team with (top) and without the new player (bottom).",
```

```
    decision_threshold = 100,
    threshold_label = "Above this threshold,\nyou win the award.",
    dimensions = c(10.26667, 6.16000))
```



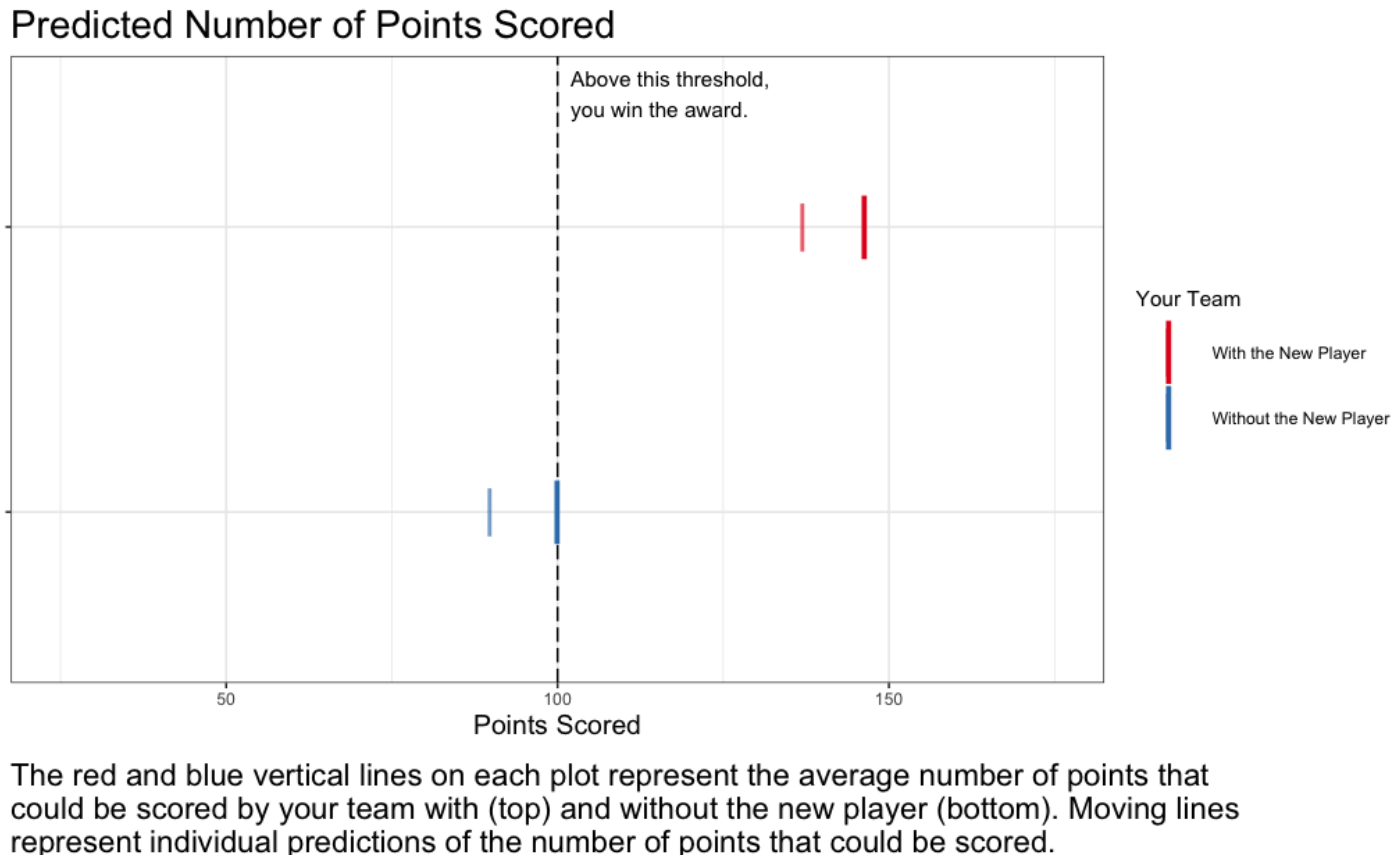
Moving lines represent individual predictions of the number of points that could be scored by your team with (top) and without the new player (bottom).

Hypothetical Outcome Plots (HOPs) with Means

A chart function for HOPs of the possible points for each version of the team, with means added.

```
hops_w_means <- function(df, n_draws, frames_per_second, data_domain, title, x_label, ca
ption, decision_threshold, threshold_label, dimensions, label_mean = FALSE) {
  plt <- df %>%
      mutate(
        quantiles = list(sample(ppoints(n_draws))), # get n_draws equally spaced quantil
es and shuffle their order
        draws = pmap(list(quantiles, mean, sd), qnorm), # map those quantiles to draws f
rom each distribution
        draw_n = if_else(Team == "With the New Player", # number draws differently for e
ach distribution so we aren't showing the same quantile for both distributions
                         list(sample(1:n_draws)),
                         list(1:n_draws))
      ) %>%
      unnest(cols = c(quantiles, draws, draw_n)) %>% # one row per draw
      ggplot(aes(y = draws, x = Team, color = Team)) +
      scale_x_discrete(limits = rev(levels(df$Team))) + # get teams in correct order (re
d on top)
      geom_hline(yintercept = decision_threshold, linetype = "longdash") + # award thres
hold
      annotate("text", y = (decision_threshold + 2), x = "With the New Player", label =
 threshold_label, hjust = 0, vjust = -2.4) +
      geom_point(shape = 124, size = HOPs_size, alpha = opacity) +
      geom_point(aes(y = mean), size = means_size_HOPs, shape = 124) + # add means
      coord_flip() +
      theme_bw() +
      scale_color_brewer(palette = "Set1") +
      ylim(data_domain[1], data_domain[2]) +
      labs(
        title = title,
        x = NULL,
        y = x_label,
        color = "Your Team",
        caption = wrap_label(caption, char_before_wrap)) +
      theme(
        axis.title = element_text(size=label_size),
        axis.text.y = element_blank(),
        plot.title = element_text(size = title_size),
        plot.caption = element_text(size = caption_size, hjust = 0, vjust = -1)) +
      transition_manual(draw_n)

  if (label_mean) {
    # get mean positions for each dists
    mean_with <- df %>% filter(Team == "With the New Player") %>% select(mean) %>% as.nu
meric()
    mean_without <- df %>% filter(Team == "Without the New Player") %>% select(mean) %>%
as.numeric()

    plt <- plt +
      geom_segment(x = 1.9, xend = 1.1, y = mean_with, yend = 146.3535, size = 0.5, line
type = "longdash", color = "black") +
      geom_segment(x = 1.1, xend = 1.1, y = mean_without, yend = 146.3535, size = 0.5, l
inetype = "longdash", color = "black") +
      annotate("text", x = 1.1, y = 148, label = "Average\npredicted\nscore", hjust = 0,
```

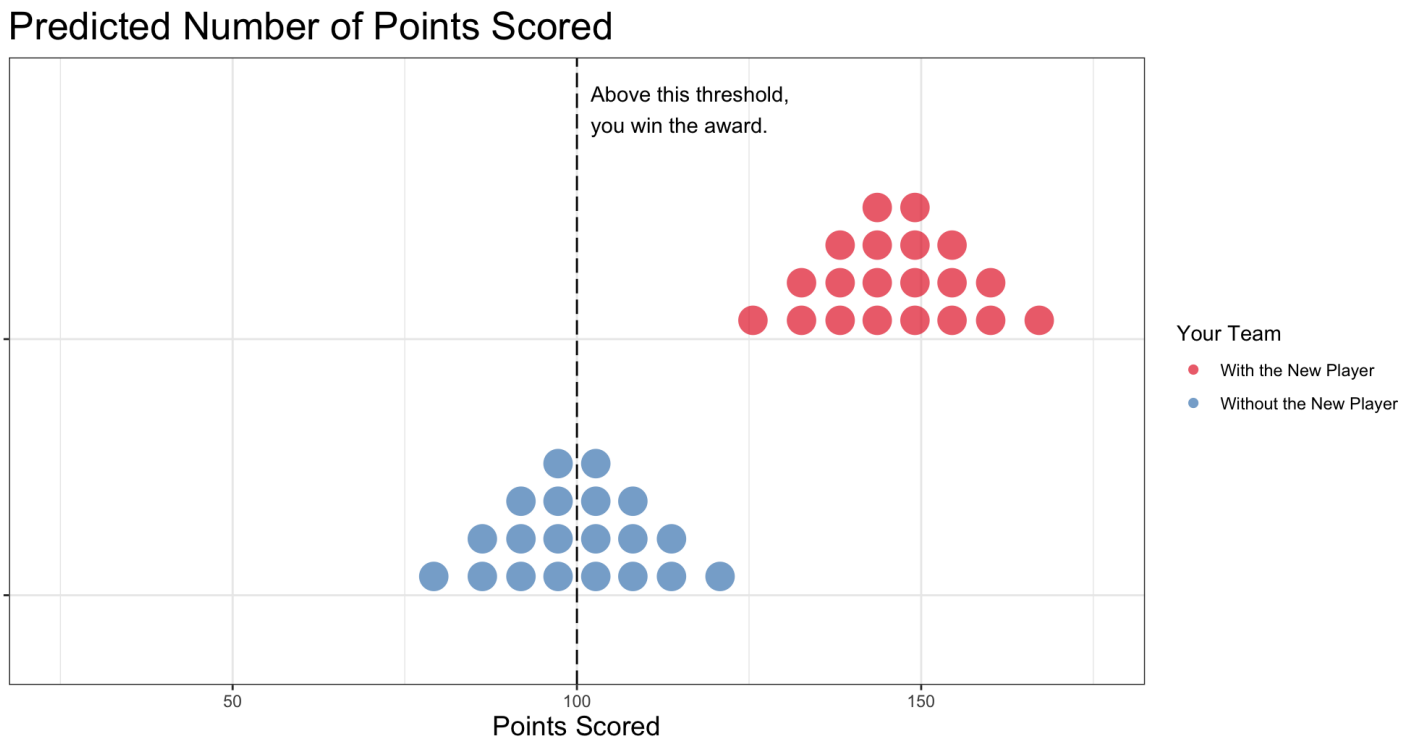
```
vjust = 0.5)
  }

  animation <- animate(plt, fps = frames_per_second, nframes = 10 * frames_per_second, r
es = 100, width = dimensions[1]*100, height = dimensions[2]*100)

  return(animation)
}

hops_w_means(df = prac_df,
             n_draws = n_draws_hops,
             frames_per_second = frame_rate,
             data_domain = data_domain,
             title = "Predicted Number of Points Scored",
             x_label = "Points Scored",
             caption = "The red and blue vertical lines on each plot represent the avera
ge number of points that could be scored by your team with (top) and without the new pla
yer (bottom). Moving lines represent individual predictions of the number of points that
could be scored.",
             decision_threshold = 100,
             threshold_label = "Above this threshold,\nyou win the award.",
             dimensions = c(10.26667, 6.16000))
```



Quantile Dotplots

A chart function for quantile dotplots of the possible points for each version of the team.

```
quantile_dotplots <- function(df, n_dots, data_domain, title, x_label, caption, decision
_threshold, threshold_label) {
  plt <- df %>%
      ggplot(aes(dist = "norm", arg1 = mean, arg2 = sd, y = rev(levels(Team)), fill = Te
am)) +
      geom_vline(xintercept = decision_threshold, linetype = "longdash") + # award thres
hold
      annotate("text", x = (decision_threshold + 2), y = "Without the New Player", label
= threshold_label, hjust = 0, vjust = -4.5) +
      stat_dist_dotsh(quantiles = n_dots, binwidth = 5, dotsize = 0.85, stackratio = 1.2
, alpha = opacity, color = NA) +
      theme_bw() +
      scale_fill_brewer(palette = "Set1") +
      coord_cartesian(
        xlim = data_domain,
        ylim = c(1.25, 2.5)) +
      labs(
        title = title,
        x = x_label,
        y = NULL,
        fill = "Your Team",
        caption = wrap_label(caption, char_before_wrap)) +
      theme(
        strip.background = element_blank(),
        strip.text.x = element_blank(),
        axis.title = element_text(size=label_size),
        axis.text.y = element_blank(),
        plot.title = element_text(size = title_size),
        plot.caption = element_text(size = caption_size, hjust = 0, vjust = -1))

  return(plt)
}

quantile_dotplots(df = prac_df,
                  n_dots = n_dots_qdps,
                  data_domain = data_domain,
                  title = "Predicted Number of Points Scored",
                  x_label = "Points Scored",
                  caption = "Each dot represents a 5% chance that different numbers of p
oints (shown along the horizontal axis) could be scored by your team with (top) and with
out the new player (bottom).",
                  decision_threshold = 100,
                  threshold_label = "Above this threshold,\nyou win the award.")
```



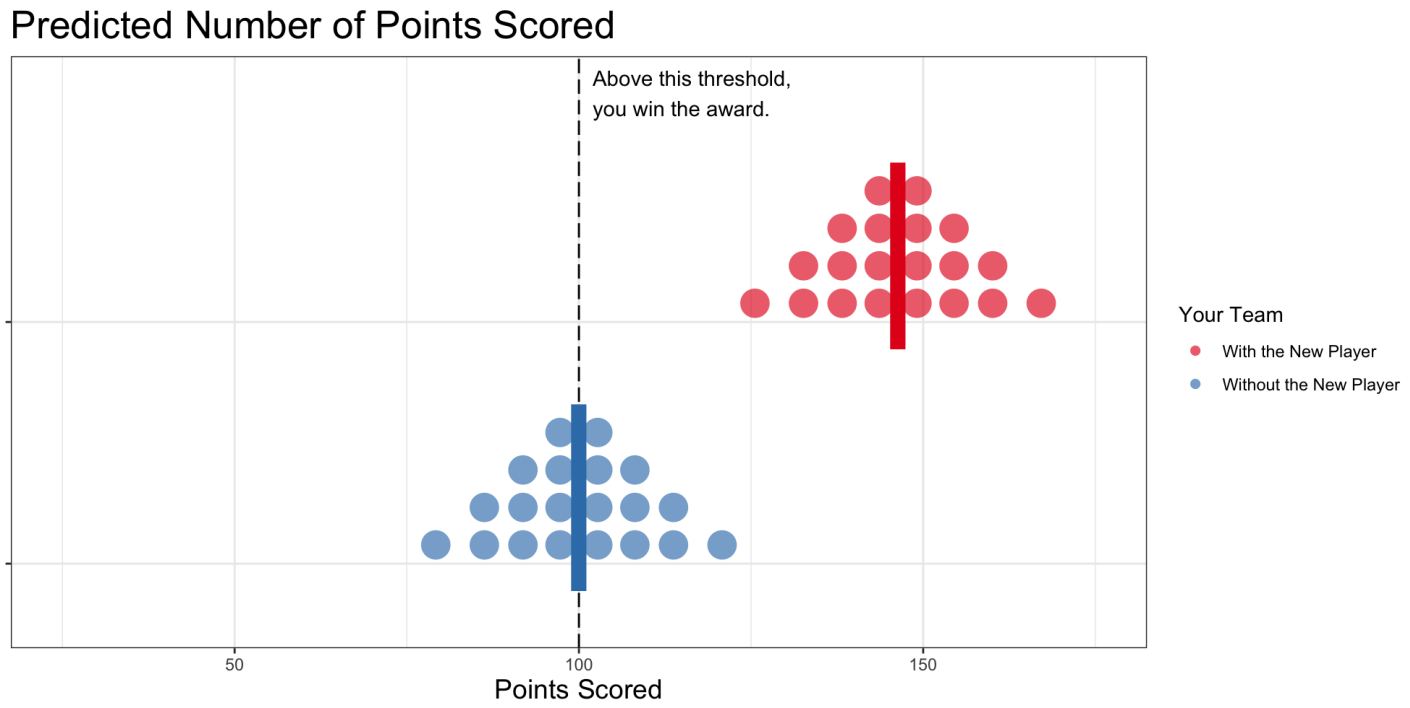
Each dot represents a 5% chance that different numbers of points (shown along the horizontal axis) could be scored by your team with (top) and without the new player (bottom).

Quantile Dotplots with Means

A chart function for quantile dotplots of the possible points for each version of the team with means added.

```
quantile_dotplots_w_means <- function(df, n_dots, data_domain, title, x_label, caption,
 decision_threshold, threshold_label, label_mean = FALSE) {
  plt <- df %>%
      ggplot(aes(dist = "norm", arg1 = mean, arg2 = sd, y = rev(levels(Team)), fill = Te
am)) +
      geom_vline(xintercept = decision_threshold, linetype = "longdash") + # award thres
hold
      annotate("text", x = (decision_threshold + 2), y = "Without the New Player", label
= threshold_label, hjust = 0, vjust = -4.5) +
      stat_dist_dotsh(quantiles = n_dots, binwidth = 5, dotsize = .85, stackratio = 1.2,
alpha = opacity, color = NA) +
      geom_point(aes(x = mean, color = Team), shape = 124, size = means_size_slab, show.
legend = FALSE, position = position_nudge(y = 0.275)) + # add means
      theme_bw() +
      scale_fill_brewer(palette = "Set1") +
      scale_color_brewer(palette = "Set1") +
      coord_cartesian(
        xlim = data_domain,
        ylim = c(1.25, 2.5)) +
      labs(
        title = title,
        x = x_label,
        y = NULL,
        fill = "Your Team",
        caption = wrap_label(caption, char_before_wrap)) +
      theme(
        strip.background = element_blank(),
        strip.text.x = element_blank(),
        axis.title = element_text(size=label_size),
        axis.text.y = element_blank(),
        plot.title = element_text(size = title_size),
        plot.caption = element_text(size = caption_size, hjust = 0, vjust = -1))

  if (label_mean) {
    # get mean positions for each dists
    mean_with <- df %>% filter(Team == "With the New Player") %>% select(mean) %>% as.nu
meric()
    mean_without <- df %>% filter(Team == "Without the New Player") %>% select(mean) %>%
as.numeric()

    plt <- plt +
      geom_segment(y = 1.95, yend = 1.6, x = mean_with, xend = 146.3535, size = 0.5, lin
etype = "longdash", color = "black") +
      geom_segment(y = 1.6, yend = 1.6, x = mean_without, xend = 146.3535, size = 0.5, l
inetype = "longdash", color = "black") +
      annotate("text", y = 1.6, x = 148, label = "Average\npredicted\nscore", hjust = 0,
vjust = 0.5)
  }

  return(plt)
}

quantile_dotplots_w_means(df = prac_df,
```

```
                          n_dots = n_dots_qdps,
                          data_domain = data_domain,
                          title = "Predicted Number of Points Scored",
                          x_label = "Points Scored",
                          caption = "The red and blue vertical lines on each plot repres
ent the average number of points that could be scored by your team with (top) and withou
t the new player (bottom). Each dot represents a 5% chance that different numbers of poi
nts (shown along the horizontal axis) could be scored.",
                          decision_threshold = 100,
                          threshold_label = "Above this threshold,\nyou win the award.")
```

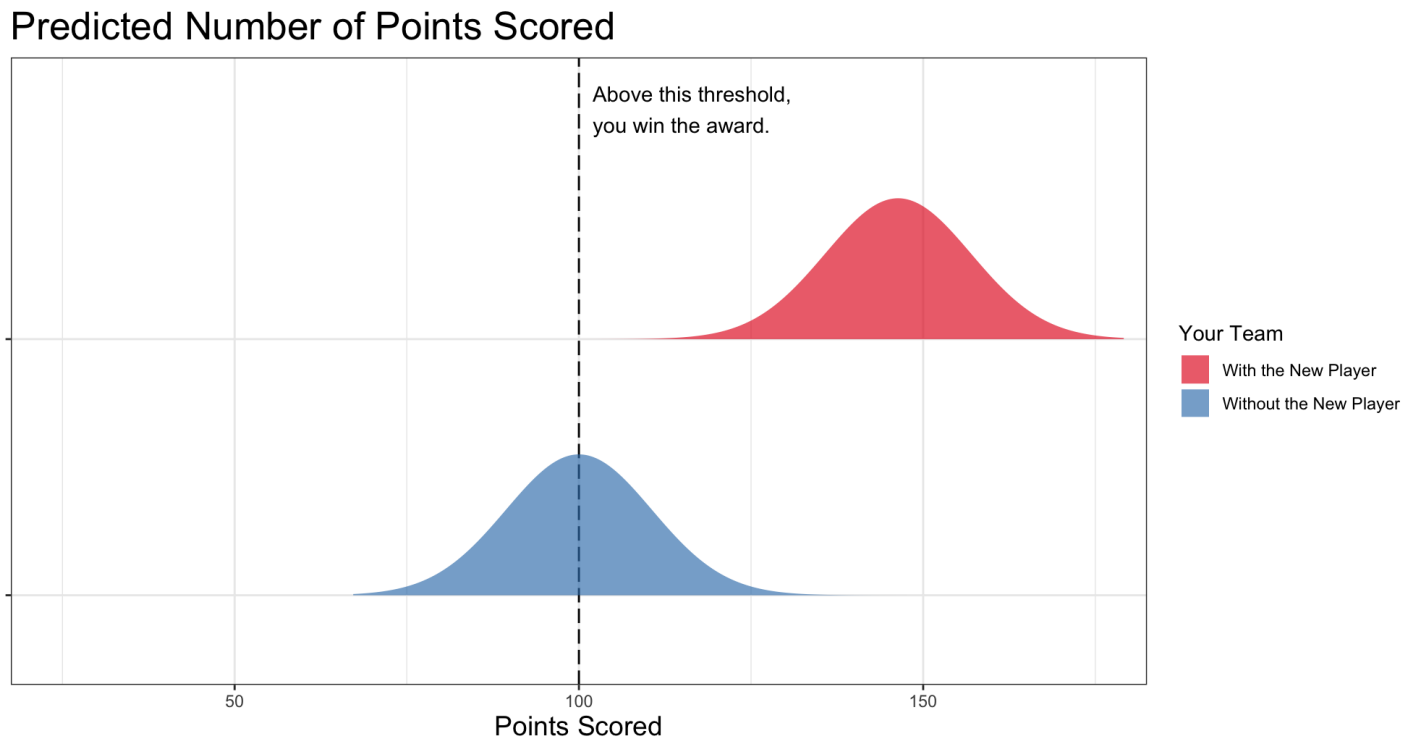


The red and blue vertical lines on each plot represent the average number of points that could be scored by your team with (top) and without the new player (bottom). Each dot represents a 5% chance that different numbers of points (shown along the horizontal axis) could be scored.

Densities

A chart function for continuous probability densities of the possible points for each version of the team.

```
densities <- function(df, data_domain, title, x_label, caption, decision_threshold, thre
shold_label) {
  plt <- df %>%
      ggplot(aes(dist = "norm", arg1 = mean, arg2 = sd, y = rev(levels(Team)), fill = Te
am)) +
      geom_vline(xintercept = decision_threshold, linetype = "longdash") + # award thres
hold
      annotate("text", x = (decision_threshold + 2), y = "Without the New Player", label
= threshold_label, hjust = 0, vjust = -4.5) +
      stat_dist_slabh(alpha = opacity, scale = 0.55) +
      theme_bw() +
      scale_fill_brewer(palette = "Set1") +
      coord_cartesian(
        xlim = data_domain,
        ylim = c(1.25, 2.5)) +
      labs(
        title = title,
        x = x_label,
        y = NULL,
        fill = "Your Team",
        caption = wrap_label(caption, char_before_wrap)) +
      theme(
        strip.background = element_blank(),
        strip.text.x = element_blank(),
        axis.title = element_text(size=label_size),
        axis.text.y = element_blank(),
        plot.title = element_text(size = title_size),
        plot.caption = element_text(size = caption_size, hjust = 0, vjust = -1))

  return(plt)
}

densities(df = prac_df,
          data_domain = data_domain,
          title = "Predicted Number of Points Scored",
          x_label = "Points Scored",
          caption = "The height of the shaded area represents the chances that different
numbers of points (shown along the horizontal axis) could be scored by your team with (t
op) and without the new player (bottom).",
          decision_threshold = 100,
          threshold_label = "Above this threshold,\nyou win the award.")
```



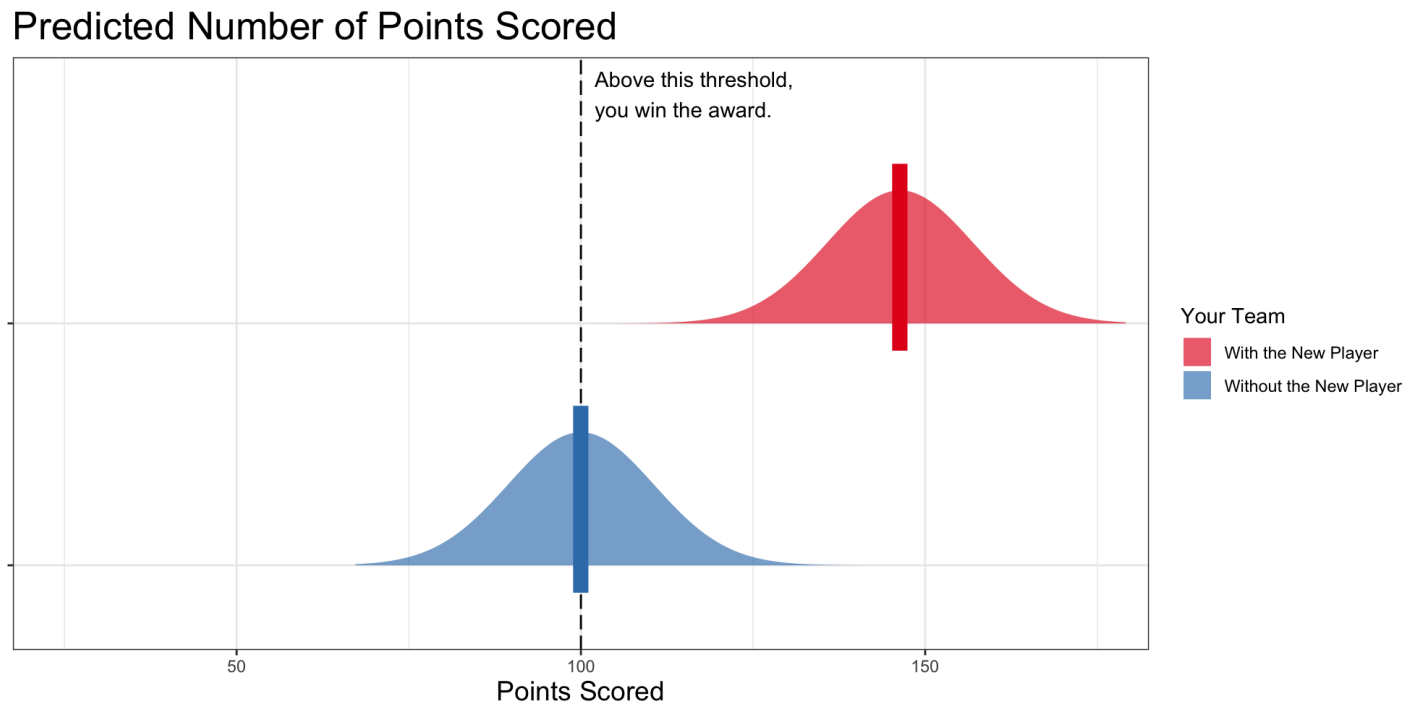
The height of the shaded area represents the chances that different numbers of points (shown along the horizontal axis) could be scored by your team with (top) and without the new player (bottom).

Densities with Means

A chart function for continuous probability densities of the possible points for each version of the team with means added.

```
densities_w_means <- function(df, data_domain, title, x_label, caption, decision_thresho
ld, threshold_label, label_mean = FALSE) {
  plt <- df %>%
      ggplot(aes(dist = "norm", arg1 = mean, arg2 = sd, y = rev(levels(Team)), fill = Te
am)) +
      geom_vline(xintercept = decision_threshold, linetype = "longdash") + # award thres
hold
      annotate("text", x = (decision_threshold + 2), y = "Without the New Player", label
= threshold_label, hjust = 0, vjust = -4.5) +
      stat_dist_slabh(alpha = opacity, scale = 0.55) +
      geom_point(aes(x = mean, color = Team), shape = 124, size = means_size_slab, show.
legend = FALSE, position = position_nudge(y = 0.275)) + # add means
      theme_bw() +
      scale_fill_brewer(palette = "Set1") +
      scale_color_brewer(palette = "Set1") +
      coord_cartesian(
        xlim = data_domain,
        ylim = c(1.25, 2.5)) +
      labs(
        title = title,
        x = x_label,
        y = NULL,
        fill = "Your Team",
        caption = wrap_label(caption, char_before_wrap)) +
      theme(
        strip.background = element_blank(),
        strip.text.x = element_blank(),
        axis.title = element_text(size=label_size),
        axis.text.y = element_blank(),
        plot.title = element_text(size = title_size),
        plot.caption = element_text(size = caption_size, hjust = 0, vjust = -1))

  if (label_mean) {
    # get mean positions for each dists
    mean_with <- df %>% filter(Team == "With the New Player") %>% select(mean) %>% as.nu
meric()
    mean_without <- df %>% filter(Team == "Without the New Player") %>% select(mean) %>%
as.numeric()

    plt <- plt +
      geom_segment(y = 1.95, yend = 1.6, x = mean_with, xend = 146.3535, size = 0.5, lin
etype = "longdash", color = "black") +
      geom_segment(y = 1.6, yend = 1.6, x = mean_without, xend = 146.3535, size = 0.5, l
inetype = "longdash", color = "black") +
      annotate("text", y = 1.6, x = 148, label = "Average\npredicted\nscore", hjust = 0,
vjust = 0.5)
  }

  return(plt)
}

densities_w_means(df = prac_df,
                  data_domain = data_domain,
```

```
                  title = "Predicted Number of Points Scored",
                  x_label = "Points Scored",
                  caption = "The red and blue vertical lines on each plot represent the
 average number of points that could be scored by your team with (top) and without the n
ew player (bottom). The height of the shaded area represents the chances that different
 numbers of points (shown along the horizontal axis) could be scored.",
                  decision_threshold = 100,
                  threshold_label = "Above this threshold,\nyou win the award.")
```



The red and blue vertical lines on each plot represent the average number of points that could be scored by your team with (top) and without the new player (bottom). The height of the shaded area represents the chances that different numbers of points (shown along the horizontal axis) could be scored.

Stimuli Generation

We create one of each chart type for each data condition above and save to a folder called stimuli.

```
# cycle through rows in the table of data conditions
for (c in unique(conds_df$condition)) {
  # isolaten data for the current condtion
  use_df <- conds_df %>% filter(condition %in% c)

  # intervals
  plt <- intervals(df = use_df,
    data_domain = data_domain,
    title = "Predicted Number of Points Scored",
    x_label = "Points Scored",
    caption = "Intervals contain 95% of the possible numbers of points that could be sco
red by your team with (top) and without the new player (bottom).",
    decision_threshold = 100,
    threshold_label = "Above this threshold,\nyou win the award.")
  fname <- paste("stimuli/intervals-", c,".svg", sep = "")
  ggsave(file = fname, plot = plt, width = dims[1], height = dims[2])

  # intervals with means
  plt <- intervals_w_means(df = use_df,
    data_domain = data_domain,
    title = "Predicted Number of Points Scored",
    x_label = "Points Scored",
    caption = "The red and blue vertical lines on each plot represent the average number
of points that could be scored by your team with (top) and without the new player (botto
m). Intervals contain 95% of the possible numbers of points that could be scored.",
    decision_threshold = 100,
    threshold_label = "Above this threshold,\nyou win the award.")
  fname <- paste("stimuli/intervals_w_means-", c, ".svg", sep = "")
  ggsave(file = fname, plot = plt, width = dims[1], height = dims[2])

  # hops
  plt <- hops(df = use_df,
    n_draws = n_draws_hops,
    frames_per_second = frame_rate,
    data_domain = data_domain,
    title = "Predicted Number of Points Scored",
    x_label = "Points Scored",
    caption = "Moving lines represent individual predictions of the number of points tha
t could be scored by your team with (top) and without the new player (bottom).",
    decision_threshold = 100,
    threshold_label = "Above this threshold,\nyou win the award.",
    dimensions = dims)
  fname <- paste("stimuli/HOPs-", c, ".gif", sep = "")
  anim_save(filename = fname, animation = plt)

  # hops with means
  plt <- hops_w_means(df = use_df,
    n_draws = n_draws_hops,
    frames_per_second = frame_rate,
    data_domain = data_domain,
    title = "Predicted Number of Points Scored",
    x_label = "Points Scored",
    caption = "The red and blue vertical lines on each plot represent the average number
of points that could be scored by your team with (top) and without the new player (botto
```

```
of points that could be scored by your team with (top) and without the new player (botto
m). Moving lines represent individual predictions of the number of points that could be
 scored.",
    decision_threshold = 100,
    threshold_label = "Above this threshold,\nyou win the award.",
    dimensions = dims)
  fname <- paste("stimuli/HOPs_w_means-", c, ".gif", sep = "")
  anim_save(filename = fname, animation = plt)

  # quantile dotplots
  plt <- quantile_dotplots(df = use_df,
    n_dots = n_dots_qdps,
    data_domain = data_domain,
    title = "Predicted Number of Points Scored",
    x_label = "Points Scored",
    caption = "Each dot represents a 5% chance that different numbers of points (shown a
long the horizontal axis) could be scored by your team with (top) and without the new pl
ayer (bottom).",
    decision_threshold = 100,
    threshold_label = "Above this threshold,\nyou win the award.")
  fname <- paste("stimuli/QDPs-", c, ".svg", sep = "")
  ggsave(file = fname, plot = plt, width = dims[1], height = dims[2])

  # quantile dotplots with means
  plt <- quantile_dotplots_w_means(df = use_df,
    n_dots = n_dots_qdps,
    data_domain = data_domain,
    title = "Predicted Number of Points Scored",
    x_label = "Points Scored",
    caption = "The red and blue vertical lines on each plot represent the average number
of points that could be scored by your team with (top) and without the new player (botto
m). Each dot represents a 5% chance that different numbers of points (shown along the ho
rizontal axis) could be scored.",
    decision_threshold = 100,
    threshold_label = "Above this threshold,\nyou win the award.")
  fname <- paste("stimuli/QDPs_w_means-", c, ".svg", sep = "")
  ggsave(file = fname, plot = plt, width = dims[1], height = dims[2])

  # densities
  plt <- densities(df = use_df,
    data_domain = data_domain,
    title = "Predicted Number of Points Scored",
    x_label = "Points Scored",
    caption = "The height of the shaded area represents the chances that different numbe
rs of points (shown along the horizontal axis) could be scored by your team with (top) a
nd without the new player (bottom).",
    decision_threshold = 100,
    threshold_label = "Above this threshold,\nyou win the award.")
  fname <- paste("stimuli/densities-", c, ".svg", sep = "")
  ggsave(file = fname, plot = plt, width = dims[1], height = dims[2])

  # densities with means
  plt <- densities_w_means(df = use_df,
    data_domain = data_domain,
    title = "Predicted Number of Points Scored"
```

```
    title = "Predicted Number of Points Scored",
    x_label = "Points Scored",
    caption = "The red and blue vertical lines on each plot represent the average number
of points that could be scored by your team with (top) and without the new player (botto
m). The height of the shaded area represents the chances that different numbers of point
s (shown along the horizontal axis) could be scored.",
    decision_threshold = 100,
    threshold_label = "Above this threshold,\nyou win the award.")
  fname <- paste("stimuli/densities_w_means-", c, ".svg", sep = "")
  ggsave(file = fname, plot = plt, width = dims[1], height = dims[2])
}
```

Mock and Practice Stimuli

We also create and save one of each chart type with an annotation indicating the mean for use in the mock and practice trials.

```
# select data for mock stimuli
mock_df <- conds_df %>% filter(p_superiority == 0.5)

# intervals
plt <- intervals(df = mock_df,
  data_domain = data_domain,
  title = "Predicted Number of Points Scored",
  x_label = "Points Scored",
  caption = "Intervals contain 95% of the possible numbers of points that could be score
d by your team with (top) and without the new player (bottom).",
  decision_threshold = 100,
  threshold_label = "Above this threshold,\nyou win the award.")
fname <- paste("stimuli/intervals-mock.svg", sep = "")
ggsave(file = fname, plot = plt, width = dims[1], height = dims[2])

# intervals with means
plt <- intervals_w_means(df = mock_df,
  data_domain = data_domain,
  title = "Predicted Number of Points Scored",
  x_label = "Points Scored",
  caption = "The red and blue vertical lines on each plot represent the average number o
f points that could be scored by your team with (top) and without the new player (botto
m). Intervals contain 95% of the possible numbers of points that could be scored.",
  decision_threshold = 100,
  threshold_label = "Above this threshold,\nyou win the award.",
  label_mean = TRUE)
fname <- paste("stimuli/intervals_w_means-mock.svg", sep = "")
ggsave(file = fname, plot = plt, width = dims[1], height = dims[2])

# hops
plt <- hops(df = mock_df,
  n_draws = n_draws_hops,
  frames_per_second = frame_rate,
  data_domain = data_domain,
  title = "Predicted Number of Points Scored",
  x_label = "Points Scored",
  caption = "Moving lines represent individual predictions of the number of points that
 could be scored by your team with (top) and without the new player (bottom).",
  decision_threshold = 100,
  threshold_label = "Above this threshold,\nyou win the award.",
  dimensions = dims)
fname <- paste("stimuli/HOPs-mock.gif", sep = "")
anim_save(filename = fname, animation = plt)

# hops with means
plt <- hops_w_means(df = mock_df,
  n_draws = n_draws_hops,
  frames_per_second = frame_rate,
  data_domain = data_domain,
  title = "Predicted Number of Points Scored",
  x_label = "Points Scored",
  caption = "The red and blue vertical lines on each plot represent the average number o
f points that could be scored by your team with (top) and without the new player (botto
```

```
m). Moving lines represent individual predictions of the number of points that could be
 scored.",
  decision_threshold = 100,
  threshold_label = "Above this threshold,\nyou win the award.",
  dimensions = dims,
  label_mean = TRUE)
fname <- paste("stimuli/HOPs_w_means-mock.gif", sep = "")
anim_save(filename = fname, animation = plt)

# quantile dotplots
plt <- quantile_dotplots(df = mock_df,
  n_dots = n_dots_qdps,
  data_domain = data_domain,
  title = "Predicted Number of Points Scored",
  x_label = "Points Scored",
  caption = "Each dot represents a 5% chance that different numbers of points (shown alo
ng the horizontal axis) could be scored by your team with (top) and without the new play
er (bottom).",
  decision_threshold = 100,
  threshold_label = "Above this threshold,\nyou win the award.")
fname <- paste("stimuli/QDPs-mock.svg", sep = "")
ggsave(file = fname, plot = plt, width = dims[1], height = dims[2])

# quantile dotplots with means
plt <- quantile_dotplots_w_means(df = mock_df,
  n_dots = n_dots_qdps,
  data_domain = data_domain,
  title = "Predicted Number of Points Scored",
  x_label = "Points Scored",
  caption = "The red and blue vertical lines on each plot represent the average number o
f points that could be scored by your team with (top) and without the new player (botto
m). Each dot represents a 5% chance that different numbers of points (shown along the ho
rizontal axis) could be scored.",
  decision_threshold = 100,
  threshold_label = "Above this threshold,\nyou win the award.",
  label_mean = TRUE)
fname <- paste("stimuli/QDPs_w_means-mock.svg", sep = "")
ggsave(file = fname, plot = plt, width = dims[1], height = dims[2])

# densities
plt <- densities(df = mock_df,
  data_domain = data_domain,
  title = "Predicted Number of Points Scored",
  x_label = "Points Scored",
  caption = "The height of the shaded area represents the chances that different numbers
of points (shown along the horizontal axis) could be scored by your team with (top) and
 without the new player (bottom).",
  decision_threshold = 100,
  threshold_label = "Above this threshold,\nyou win the award.")
fname <- paste("stimuli/densities-mock.svg", sep = "")
ggsave(file = fname, plot = plt, width = dims[1], height = dims[2])

# densities with means
plt <- densities_w_means(df = mock_df,
```

```
  data_domain = data_domain,
  title = "Predicted Number of Points Scored",
  x_label = "Points Scored",
  caption = "The red and blue vertical lines on each plot represent the average number o
f points that could be scored by your team with (top) and without the new player (botto
m). The height of the shaded area represents the chances that different numbers of point
s (shown along the horizontal axis) could be scored.",
  decision_threshold = 100,
  threshold_label = "Above this threshold,\nyou win the award.",
  label_mean = TRUE)
fname <- paste("stimuli/densities_w_means-mock.svg", sep = "")
ggsave(file = fname, plot = plt, width = dims[1], height = dims[2])
```

```
# select data for mock stimuli
prac_df <- conds_df %>% filter(p_superiority == 0.999)

# intervals
plt <- intervals(df = prac_df,
  data_domain = data_domain,
  title = "Predicted Number of Points Scored",
  x_label = "Points Scored",
  caption = "Intervals contain 95% of the possible numbers of points that could be score
d by your team with (top) and without the new player (bottom).",
  decision_threshold = 100,
  threshold_label = "Above this threshold,\nyou win the award.")
fname <- paste("stimuli/intervals-prac.svg", sep = "")
ggsave(file = fname, plot = plt, width = dims[1], height = dims[2])

# intervals with means
plt <- intervals_w_means(df = prac_df,
  data_domain = data_domain,
  title = "Predicted Number of Points Scored",
  x_label = "Points Scored",
  caption = "The red and blue vertical lines on each plot represent the average number o
f points that could be scored by your team with (top) and without the new player (botto
m). Intervals contain 95% of the possible numbers of points that could be scored.",
  decision_threshold = 100,
  threshold_label = "Above this threshold,\nyou win the award.",
  label_mean = TRUE)
fname <- paste("stimuli/intervals_w_means-prac.svg", sep = "")
ggsave(file = fname, plot = plt, width = dims[1], height = dims[2])

# hops
plt <- hops(df = prac_df,
  n_draws = n_draws_hops,
  frames_per_second = frame_rate,
  data_domain = data_domain,
  title = "Predicted Number of Points Scored",
  x_label = "Points Scored",
  caption = "Moving lines represent individual predictions of the number of points that
 could be scored by your team with (top) and without the new player (bottom).",
  decision_threshold = 100,
  threshold_label = "Above this threshold,\nyou win the award.",
  dimensions = dims)
fname <- paste("stimuli/HOPs-prac.gif", sep = "")
anim_save(filename = fname, animation = plt)

# hops with means
plt <- hops_w_means(df = prac_df,
  n_draws = n_draws_hops,
  frames_per_second = frame_rate,
  data_domain = data_domain,
  title = "Predicted Number of Points Scored",
  x_label = "Points Scored",
  caption = "The red and blue vertical lines on each plot represent the average number o
f points that could be scored by your team with (top) and without the new player (botto
```

```
m). Moving lines represent individual predictions of the number of points that could be
 scored.",
  decision_threshold = 100,
  threshold_label = "Above this threshold,\nyou win the award.",
  dimensions = dims,
  label_mean = TRUE)
fname <- paste("stimuli/HOPs_w_means-prac.gif", sep = "")
anim_save(filename = fname, animation = plt)

# quantile dotplots
plt <- quantile_dotplots(df = prac_df,
  n_dots = n_dots_qdps,
  data_domain = data_domain,
  title = "Predicted Number of Points Scored",
  x_label = "Points Scored",
  caption = "Each dot represents a 5% chance that different numbers of points (shown alo
ng the horizontal axis) could be scored by your team with (top) and without the new play
er (bottom).",
  decision_threshold = 100,
  threshold_label = "Above this threshold,\nyou win the award.")
fname <- paste("stimuli/QDPs-prac.svg", sep = "")
ggsave(file = fname, plot = plt, width = dims[1], height = dims[2])

# quantile dotplots with means
plt <- quantile_dotplots_w_means(df = prac_df,
  n_dots = n_dots_qdps,
  data_domain = data_domain,
  title = "Predicted Number of Points Scored",
  x_label = "Points Scored",
  caption = "The red and blue vertical lines on each plot represent the average number o
f points that could be scored by your team with (top) and without the new player (botto
m). Each dot represents a 5% chance that different numbers of points (shown along the ho
rizontal axis) could be scored.",
  decision_threshold = 100,
  threshold_label = "Above this threshold,\nyou win the award.",
  label_mean = TRUE)
fname <- paste("stimuli/QDPs_w_means-prac.svg", sep = "")
ggsave(file = fname, plot = plt, width = dims[1], height = dims[2])

# densities
plt <- densities(df = prac_df,
  data_domain = data_domain,
  title = "Predicted Number of Points Scored",
  x_label = "Points Scored",
  caption = "The height of the shaded area represents the chances that different numbers
of points (shown along the horizontal axis) could be scored by your team with (top) and
 without the new player (bottom).",
  decision_threshold = 100,
  threshold_label = "Above this threshold,\nyou win the award.")
fname <- paste("stimuli/densities-prac.svg", sep = "")
ggsave(file = fname, plot = plt, width = dims[1], height = dims[2])

# densities with means
plt <- densities_w_means(df = prac_df,
```

```
  data_domain = data_domain,
  title = "Predicted Number of Points Scored",
  x_label = "Points Scored",
  caption = "The red and blue vertical lines on each plot represent the average number o
f points that could be scored by your team with (top) and without the new player (botto
m). The height of the shaded area represents the chances that different numbers of point
s (shown along the horizontal axis) could be scored.",
  decision_threshold = 100,
  threshold_label = "Above this threshold,\nyou win the award.",
  label_mean = TRUE)
fname <- paste("stimuli/densities_w_means-prac.svg", sep = "")
ggsave(file = fname, plot = plt, width = dims[1], height = dims[2])
```