StimuliGeneration

Alex Kale

6/18/2019

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

Data Conditions

We manipulate the probability of the new machine producing more widgets than the old machine (p_superiority), sampling at 24 linear spaced intervals in logodds units.

```
# linear sampling of log odds for ground truth probability of superiority for the new machine
logodds <- seq(log(0.025/(1-0.025)), log(0.975/(1-0.975)), length.out = 24)
p_sup <- 1 / (1 + exp(-logodds))

print(p_sup)
```

```
##  [1] 0.02500000 0.03405957 0.04624645 0.06251174 0.08399386 0.11197654
##  [7] 0.14777771 0.19254342 0.24694046 0.31079046 0.38275856 0.46026266
## [13] 0.53973734 0.61724144 0.68920954 0.75305954 0.80745658 0.85222229
## [19] 0.88802346 0.91600614 0.93748826 0.95375355 0.96594043 0.97500000
```

We also manipulate the baseline probability of gaining/keeping the contract with the old machine. We sample three levels of this baseline probability: 0.5 where the old machine is as likely as a coin flip to result in the contract, 0.15 where the old machine is fairly unlikely to result in the contract, and 0.85 where the old machine has a good chance of resulting in the contract.

```
# baseline probability of gaining/keeping a contract with the old machine
base <- c(.15, .5, .85)

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

head(conds_df)
```

```
##   p_superiority baseline
## 1    0.02500000     0.15
## 2    0.03405957     0.15
## 3    0.04624645     0.15
## 4    0.06251174     0.15
## 5    0.08399386     0.15
## 6    0.11197654     0.15
```

We also want to creat stimuli for the practice trials. To make these trials easy, we choose a baseline probability of gaining/keeping the contract of 0.95 and probability of superiority values near 0.5. This way it should be obvious that the new machine is not worth the cost, and we can use these trials as an attention check. We create a gain framing version where probability of superiority is 0.6 and a loss framing version where probability of superiority is 0.4.

```
# create df containting rows for practice trials
prac_df <- data.frame(
  "p_superiority" = c(.4, .6),
  "baseline" = c(.95))
# append to conditions dataframe
conds_df <- rbind(conds_df, prac_df)

head(conds_df)
```

```
##   p_superiority baseline
## 1    0.02500000     0.15
## 2    0.03405957     0.15
## 3    0.04624645     0.15
## 4    0.06251174     0.15
## 5    0.08399386     0.15
## 6    0.11197654     0.15
```

When p_superiority is greater than 0.5, the decision task is framed as a gain scenario where the user needs to manufacture at least 500 million widgets next year to get a new contract. When p_superiority is less than 0.5, the decision task is framed as a loss scenario where the user needs to manufacture no more than 75 defective widgets per million next year to keep an existing contract.

```
# label gain vs loss framing trials based on p_superiority and add contract threshold
s
conds_df <- conds_df %>%
  mutate(frame = if_else(p_superiority > .5, "gain", "loss"),
         threshold = if_else(frame=="gain",
                             500, # million widgets required to gain contract
                             75)) # defective widgets per million required to keep co
ntract

head(conds_df)
```

```
##   p_superiority baseline frame threshold
## 1    0.02500000     0.15  loss        75
## 2    0.03405957     0.15  loss        75
## 3    0.04624645     0.15  loss        75
## 4    0.06251174     0.15  loss        75
## 5    0.08399386     0.15  loss        75
## 6    0.11197654     0.15  loss        75
```

We control the standard deviation of the distribution of the difference in widgets between the two machines (sd_diff) by setting it to 15. In the gain framing this is 15 million widgets. In the loss framing, this is 15 defective widgets per million. Since the value of sd_diff is relative to the threshold for gaining/keeping the contract, we can think of this variable as constant across trials. We then derive the mean difference in the number of widgets produced by the new minus the old machine (mean_diff) from sd_diff and p_superiority.

```
# add columns for the mean and standard deviation of the difference in the number of
 widgets produced by the new vs old machine
# depending on the gain vs loss frame, these values represent millions of widgets vs
 defective widgets per million
conds_df <- conds_df %>%
  mutate(sd_diff = 15, # std(new - old)
         mean_diff = sd_diff * qnorm(p_superiority)) # mean(new - old)

head(conds_df)
```

```
##   p_superiority baseline frame threshold sd_diff mean_diff
## 1    0.02500000     0.15  loss        75      15 -29.39946
## 2    0.03405957     0.15  loss        75      15 -27.36327
## 3    0.04624645     0.15  loss        75      15 -25.23588
## 4    0.06251174     0.15  loss        75      15 -23.01038
## 5    0.08399386     0.15  loss        75      15 -20.68048
## 6    0.11197654     0.15  loss        75      15 -18.24125
```

Now we calculate the summary statistics for each machine, making the dataframe double its length up to this pointWe derive the standard deviation of the number of widgets produced by the machines from year to year (sd) from sd_diff, variance sum law, and the assumption that the machines have equal and independent variances. We derive the mean number of widgets produced by each machine (mean) from the threshold for gaining/keeping the contract, the sd of widgets for each machine, and the mean_diff between the number of widgets for the new minus the old machine. We derive the probability of gaining/keeping the contract from the threshold, mean, and sd.

```
# double the length of the dataframe to add information per machine, creating a stimu
lus dataframe with a row per distribution to visualize
conds_df <- map_df(seq_len(2), ~conds_df)
conds_df$Machine <- sort(rep(c("New", "Old"), length(conds_df$p_superiority)/2))

# add columns for the mean and standard deviation of widgets for each machine and the
probability of gaining/keeping the contract
conds_df <- conds_df %>%
  mutate(sd = sqrt(conds_df$sd_diff ^ 2 / 2), # assume equal and independent variance
s in the number of widgets produced by each machine
        mean = if_else(Machine=="Old",
                       if_else(frame=="gain", # old machine is at baseline
                               threshold - sd * qnorm(1 - baseline),
                               threshold - sd * qnorm(baseline)),
                       if_else(frame=="gain", # new machine is at difference from bas
eline
                               threshold - sd * qnorm(1 - baseline) + mean_diff,
                               threshold - sd * qnorm(baseline) + mean_diff)),
        p_contract = if_else(frame=="gain", # probability of exceeding threshold to g
ain/keep contract
                              1 - pnorm((threshold - mean)/sd),
                              pnorm((threshold - mean)/sd)))

head(conds_df)
```

```
##   p_superiority baseline frame threshold sd_diff mean_diff Machine      sd
## 1    0.02500000     0.15  loss        75      15 -29.39946     New 10.6066
## 2    0.03405957     0.15  loss        75      15 -27.36327     New 10.6066
## 3    0.04624645     0.15  loss        75      15 -25.23588     New 10.6066
## 4    0.06251174     0.15  loss        75      15 -23.01038     New 10.6066
## 5    0.08399386     0.15  loss        75      15 -20.68048     New 10.6066
## 6    0.11197654     0.15  loss        75      15 -18.24125     New 10.6066
##       mean p_contract
## 1 56.59358  0.9586627
## 2 58.62977  0.9386332
## 3 60.75716  0.9103361
## 4 62.98266  0.8713941
## 5 65.31256  0.8194683
## 6 67.75178  0.7528130
```

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"), col
lapse = "_")) %>%
  ungroup() # need to undo rowwise

head(conds_df)
```

```
## # A tibble: 6 x 11
##   p_superiority baseline frame threshold sd_diff mean_diff Machine    sd
##           <dbl>    <dbl> <chr>     <dbl>   <dbl>     <dbl> <chr>   <dbl>
## 1        0.025      0.15 loss         75      15     -29.4 New      10.6
## 2        0.0341     0.15 loss         75      15     -27.4 New      10.6
## 3        0.0462     0.15 loss         75      15     -25.2 New      10.6
## 4        0.0625     0.15 loss         75      15     -23.0 New      10.6
## 5        0.0840     0.15 loss         75      15     -20.7 New      10.6
## 6        0.112      0.15 loss         75      15     -18.2 New      10.6
## # ... with 3 more variables: mean <dbl>, p_contract <dbl>, condition <chr>
```

Since HOPs, quantile dotplots, and densities rely on samples from the underlying data generating process, we need to generate those. However, we don’t want to change the shape of our dataframe, so we nest these samples (i.e., draws) inside lists and will later unnest them as needed to produce these specific charts.

```
# for HOPs and quantile dotplots we need to add draws to our dataframe
n <- 1000 # number of samples
n_dots <- 20 # number of dots for quantile dotplots
conds_df$sample_n <- n
conds_df <- conds_df %>% as.tibble() %>%
  mutate(draws = pmap(list(sample_n, mean, sd), rnorm), # get a list of draws from th
e distribution for each condition
         draw_n = list(seq(1, n)), # number each sample in order to animate multiple
 views simultaneously
         quantiles = map(draws, ~ quantile(unlist(.x), ppoints(n_dots)))) # use draw
 to get quantiles
         # leave these draws and quantiles nested in the dataframe for later use sinc
e they are not relevant to most visualizations

head(conds_df)
```

```
## # A tibble: 6 x 15
##   p_superiority baseline frame threshold sd_diff mean_diff Machine    sd
##           <dbl>    <dbl> <chr>     <dbl>   <dbl>     <dbl> <chr>   <dbl>
## 1        0.025      0.15 loss         75      15     -29.4 New      10.6
## 2        0.0341     0.15 loss         75      15     -27.4 New      10.6
## 3        0.0462     0.15 loss         75      15     -25.2 New      10.6
## 4        0.0625     0.15 loss         75      15     -23.0 New      10.6
## 5        0.0840     0.15 loss         75      15     -20.7 New      10.6
## 6        0.112      0.15 loss         75      15     -18.2 New      10.6
## # ... with 7 more variables: mean <dbl>, p_contract <dbl>,
## #   condition <chr>, sample_n <dbl>, draws <list>, draw_n <list>,
## #   quantiles <list>
```

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 gain framing 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
gain_prac_df <- conds_df %>% filter(p_superiority == 0.6)

head(gain_prac_df)
```

```
## # A tibble: 2 x 15
##   p_superiority baseline frame threshold sd_diff mean_diff Machine    sd
##           <dbl>    <dbl> <chr>     <dbl>   <dbl>     <dbl> <chr>   <dbl>
## 1           0.6     0.95 gain        500      15      3.80 New      10.6
## 2           0.6     0.95 gain        500      15      3.80 Old      10.6
## # ... with 7 more variables: mean <dbl>, p_contract <dbl>,
## #   condition <chr>, sample_n <dbl>, draws <list>, draw_n <list>,
## #   quantiles <list>
```

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 separate x-axis domains for gain/loss framing, 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_gain <- c(445, 565)
data_domain_loss <- c(0, 135)

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

# geom sizes
means_size <- 3
HOPs_size <- 10
HOPs_mean_size_factor <- 1.5
interval_mean_size_factor <- 1.85

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

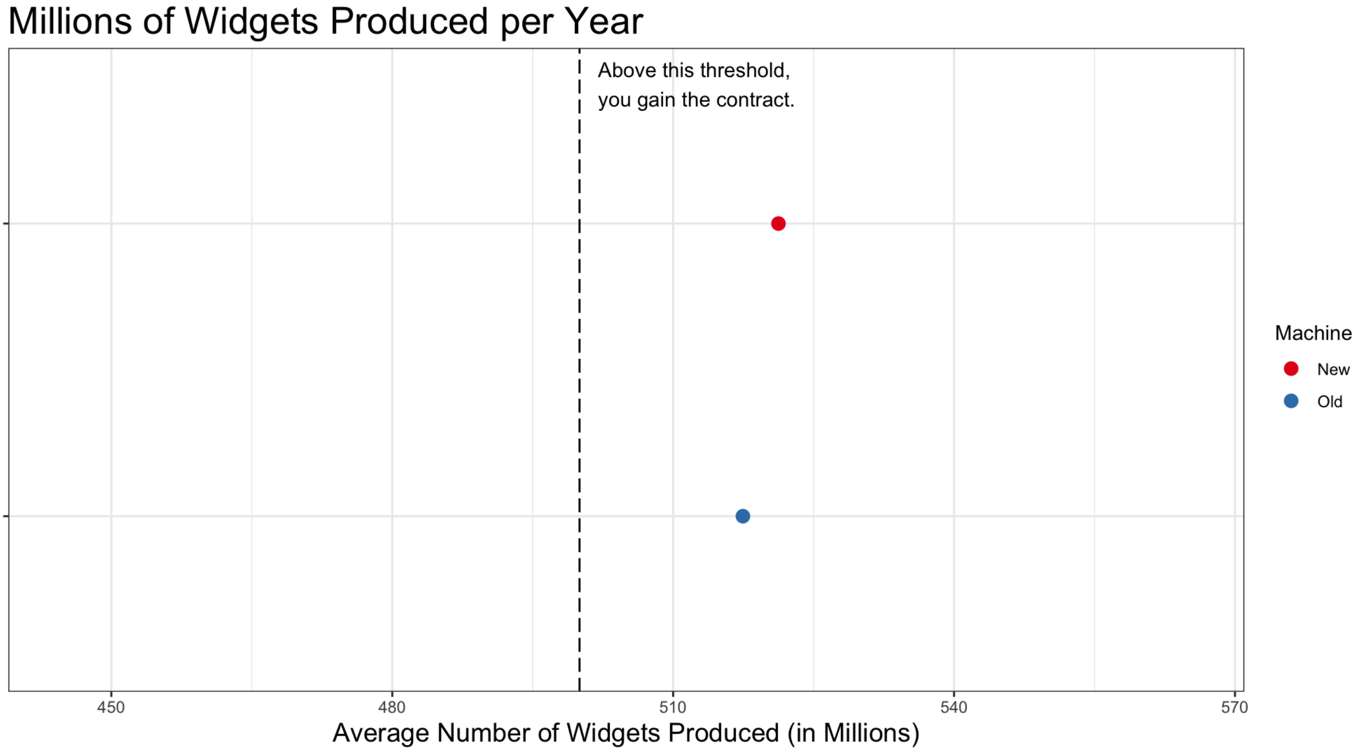
Means Only

A chart function for visualizations showing only means.

```
means_only <- function(df, data_domain, title, x_label, caption, decision_threshold,
 threshold_label) {
  plt <- df %>% ggplot(aes(x = mean, y = reorder(Machine, desc(Machine)), color = Mac
hine)) +
      geom_point(size = means_size) +
      theme_bw() +
      scale_color_brewer(palette = "Set1") +
      xlim(data_domain[1], data_domain[2]) +
      labs(
        title = title,
        x = x_label,
        y = NULL,
        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)) +
      geom_vline(xintercept = decision_threshold, linetype = "longdash") + # contract
threshold
      annotate("text", x = (decision_threshold + 2), y = 2.4, label = threshold_labe
l, hjust = 0, vjust = 0)

  return(plt)
}

means_only(df = gain_prac_df,
           data_domain = data_domain_gain,
           title = "Millions of Widgets Produced per Year",
           x_label = "Average Number of Widgets Produced (in Millions)",
           caption = "Dots represent the average number of widgets (in millions) that
could be produced by each machine in a given year.",
           decision_threshold = 500,
           threshold_label = "Above this threshold,\nyou gain the contract.")
```



Dots represent the average number of widgets (in millions) that could be produced by each machine in a given year.

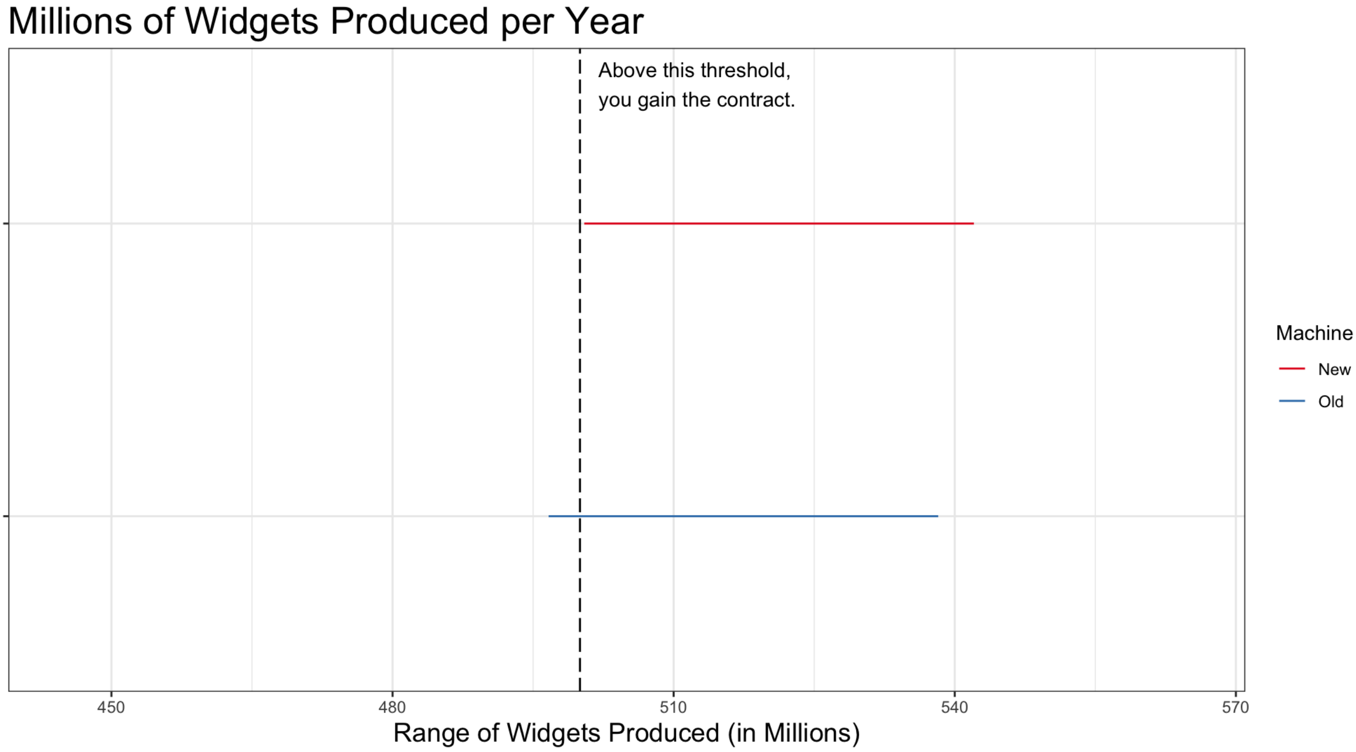
Intervals Only

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

```
intervals_only <- function(df, data_domain, title, x_label, caption, decision_thresho
ld, threshold_label) {
  plt <- df %>% ggplot(aes(y = mean, x = reorder(Machine, desc(Machine)), color = Mac
hine)) +
      geom_errorbar(aes(ymin = mean + qnorm(0.025) * sd, ymax = mean + qnorm(0.975) *
sd, width = 0)) +
      coord_flip() +
      theme_bw() +
      scale_color_brewer(palette = "Set1") +
      ylim(data_domain[1], data_domain[2]) +
      labs(
        title = title,
        x = NULL,
        y = x_label,
        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)) +
      geom_hline(yintercept = decision_threshold, linetype = "longdash") + # contract
threshold
      annotate("text", y = (decision_threshold + 2), x = 2.4, label = threshold_labe
l, hjust = 0, vjust = 0)

  return(plt)
}

intervals_only(df = gain_prac_df,
               data_domain = data_domain_gain,
               title = "Millions of Widgets Produced per Year",
               x_label = "Range of Widgets Produced (in Millions)",
               caption = "Intervals contain 95% of the possible numbers of widgets (i
n millions) that could be produced by each machine in a given year.",
               decision_threshold = 500,
               threshold_label = "Above this threshold,\nyou gain the contract.")
```



Intervals contain 95% of the possible numbers of widgets (in millions) that could be produced by each machine in a given year.

Intervals With Means

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

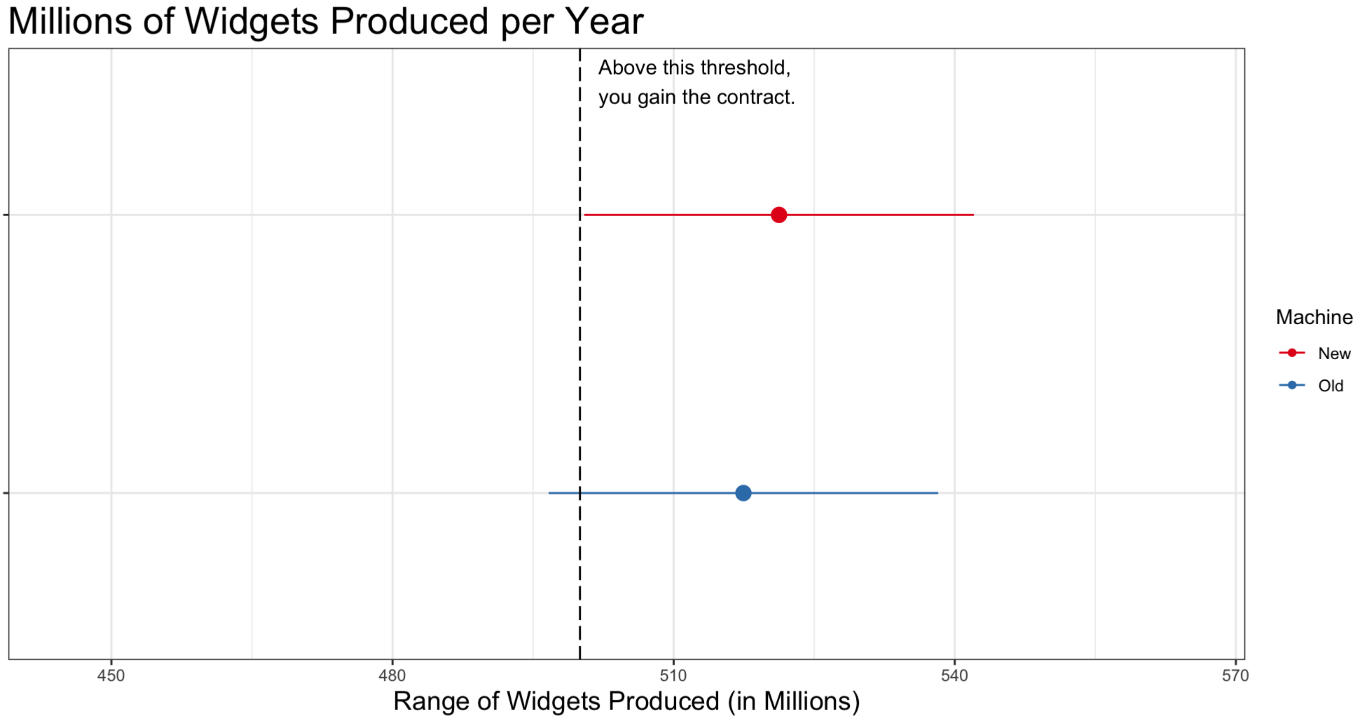
```
intervals_w_means <- function(df, data_domain, title, x_label, caption, decision_thre
shold, threshold_label) {
  plt <- df %>% ggplot(aes(y = mean, x = reorder(Machine, desc(Machine)), color = Mac
hine)) +
      geom_pointrange(aes(ymin = mean + qnorm(0.025) * sd, ymax = mean + qnorm(0.975)
* sd), show.legend = FALSE, fatten = means_size * interval_mean_size_factor) +
      geom_line(aes(y = mean - 1000)) + geom_point(aes(y = mean - 1000)) + # 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,
        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)) +
      geom_hline(yintercept = decision_threshold, linetype = "longdash") + # contract
threshold
      annotate("text", y = (decision_threshold + 2), x = 2.4, label = threshold_labe
l, hjust = 0, vjust = 0)

  return(plt)
}

intervals_w_means(df = gain_prac_df,
                  data_domain = data_domain_gain,
                  title = "Millions of Widgets Produced per Year",
                  x_label = "Range of Widgets Produced (in Millions)",
                  caption = "Dots represent the average number of widgets (in million
s) that could be produced by each machine in a given year. Intervals contain 95% of t
he possible numbers of widgets that could be produced in a given year.",
                  decision_threshold = 500,
                  threshold_label = "Above this threshold,\nyou gain the contract.")
```

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

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



Dots represent the average number of widgets (in millions) that could be produced by each machine in a given year. Intervals contain 95% of the possible numbers of widgets that could be produced in a given year.

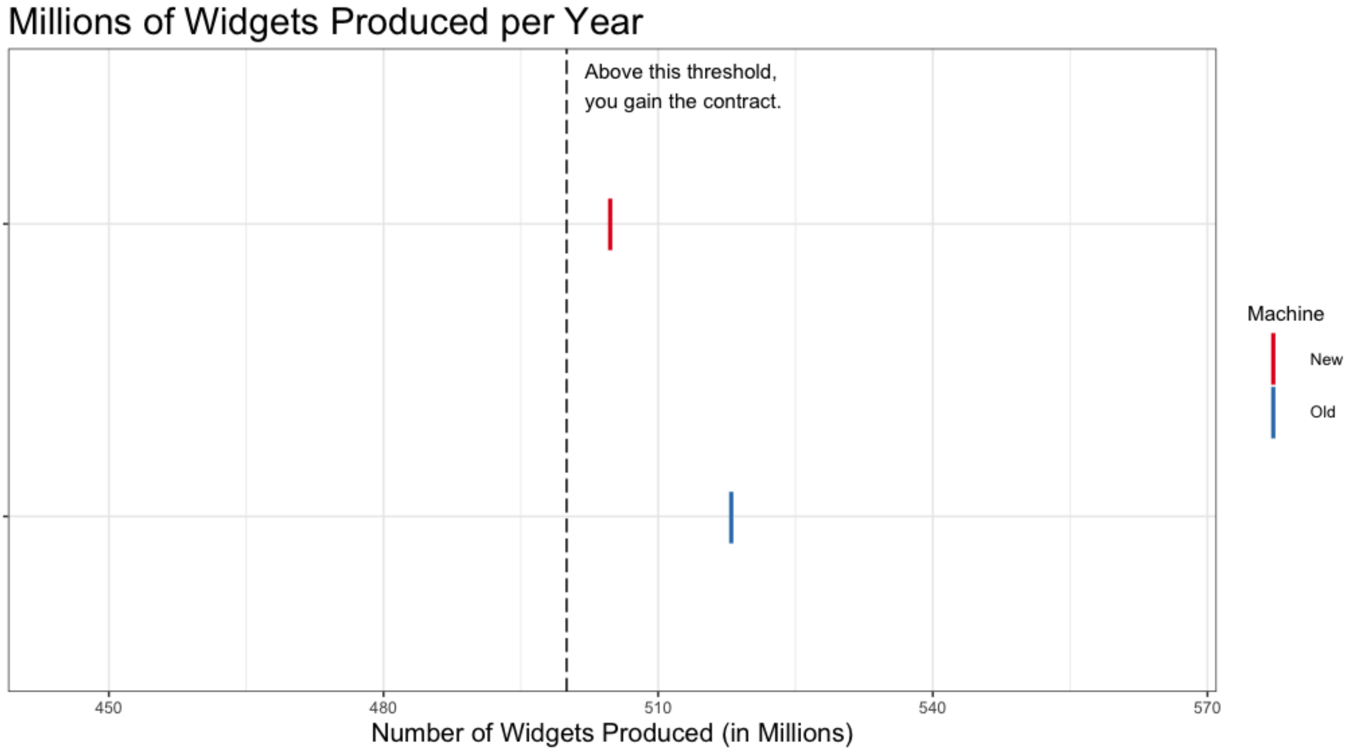
Hypothetical Outcome Plots (HOPs)

A chart function for HOPs of the possible output for each machine.

```
hops <- function(df, n_draws, frames_per_second, data_domain, title, x_label, captio
n, decision_threshold, threshold_label, dimensions) {
  plt <- df %>% select(-one_of(c("quantiles"))) %>% unnest() %>%
      filter(draw_n %in% 1:n_draws) %>% # filter to set number of draws
      ggplot(aes(y = draws, x = reorder(Machine, desc(Machine)), color = Machine)) +
      geom_point(shape = 124, size = HOPs_size) +
      coord_flip() +
      theme_bw() +
      scale_color_brewer(palette = "Set1") +
      ylim(data_domain[1], data_domain[2]) +
      labs(
        title = title,
        x = NULL,
        y = x_label,
        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)) +
      geom_hline(yintercept = decision_threshold, linetype = "longdash") + # contract
threshold
      annotate("text", y = (decision_threshold + 2), x = 2.4, label = threshold_labe
l, hjust = 0, vjust = 0) +
      transition_manual(draw_n)
    animation <- animate(plt, fps = frames_per_second, nframes = 10 * frames_per_seco
nd, res = 100, width = dimensions[1]*100, height = dimensions[2]*100)

  return(animation)
}

hops(df = gain_prac_df,
     n_draws = n_draws_hops,
     frames_per_second = frame_rate,
     data_domain = data_domain_gain,
     title = "Millions of Widgets Produced per Year",
     x_label = "Number of Widgets Produced (in Millions)",
     caption = "Lines represent the number of widgets (in millions) that could be pro
duced by each machine in a given year.",
     decision_threshold = 500,
     threshold_label = "Above this threshold,\nyou gain the contract.",
     dimensions = c(10.26667, 6.16000))
```



Lines represent the number of widgets (in millions) that could be produced by each machine in a given year.

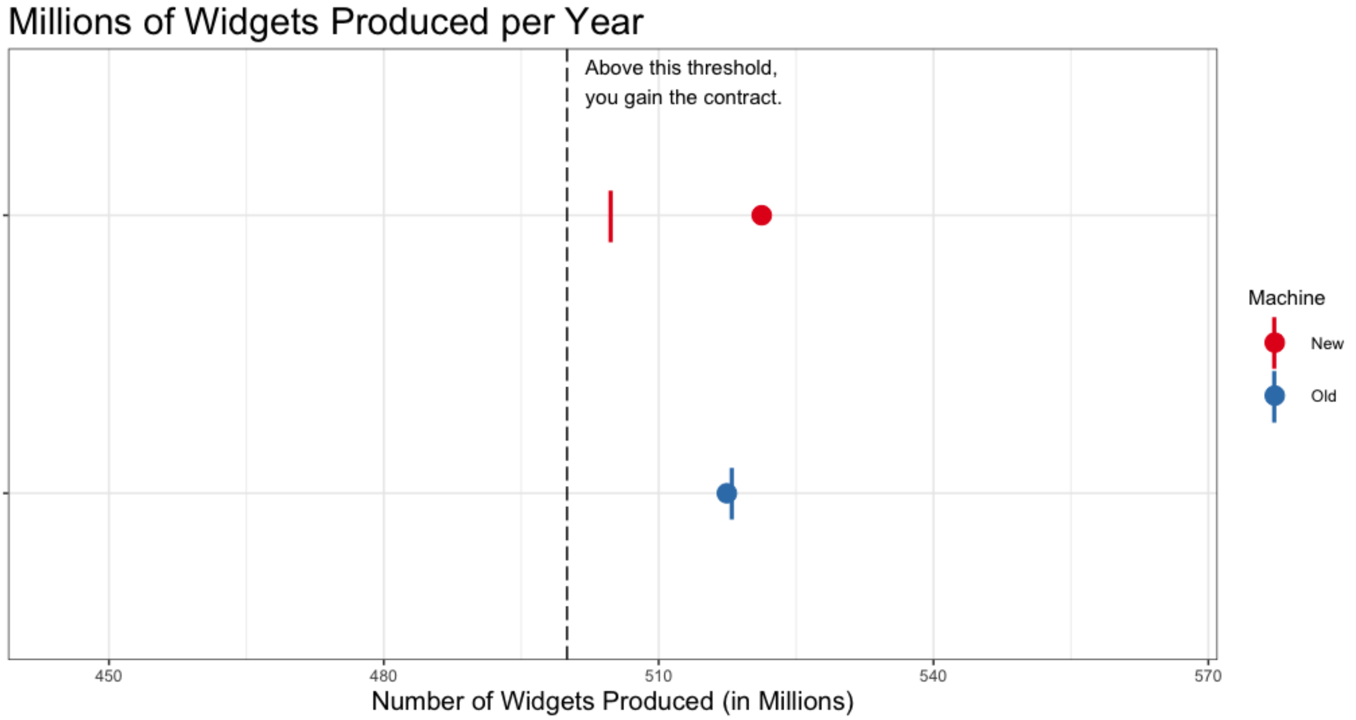
Hypothetical Outcome Plots (HOPs) with Means

A chart function for HOPs of the possible output for each machine with means.

```
hops_w_means <- function(df, n_draws, frames_per_second, data_domain, title, x_label,
caption, decision_threshold, threshold_label, dimensions) {
  plt <- df %>% select(-one_of(c("quantiles"))) %>% unnest() %>%
      filter(draw_n %in% 1:n_draws) %>% # filter to set number of draws
      ggplot(aes(y = mean, x = reorder(Machine, desc(Machine)), color = Machine)) +
      geom_point(size = means_size * HOPs_mean_size_factor) +
      geom_point(aes(y = draws), shape = 124, size = HOPs_size) +
      coord_flip() +
      theme_bw() +
      scale_color_brewer(palette = "Set1") +
      ylim(data_domain[1], data_domain[2]) +
      labs(
        title = title,
        x = NULL,
        y = x_label,
        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)) +
      geom_hline(yintercept = decision_threshold, linetype = "longdash") + # contract
threshold
      annotate("text", y = (decision_threshold + 2), x = 2.4, label = threshold_labe
l, hjust = 0, vjust = 0) +
      transition_manual(draw_n)
    animation <- animate(plt, fps = frames_per_second, nframes = 10 * frames_per_seco
nd, res = 100, width = dimensions[1]*100, height = dimensions[2]*100)

  return(animation)
}

hops_w_means(df = gain_prac_df,
             n_draws = n_draws_hops,
             frames_per_second = frame_rate,
             data_domain = data_domain_gain,
             title = "Millions of Widgets Produced per Year",
             x_label = "Number of Widgets Produced (in Millions)",
             caption = "Dots represent the average number of widgets (in millions) th
at could be produced by each machine in a given year. Lines represent the number of w
idgets that could produced in a given year.",
             decision_threshold = 500,
             threshold_label = "Above this threshold,\nyou gain the contract.",
             dimensions = c(10.26667, 6.16000))
```



Dots represent the average number of widgets (in millions) that could be produced by each machine in a given year. Lines represent the number of widgets that could produced in a given year.

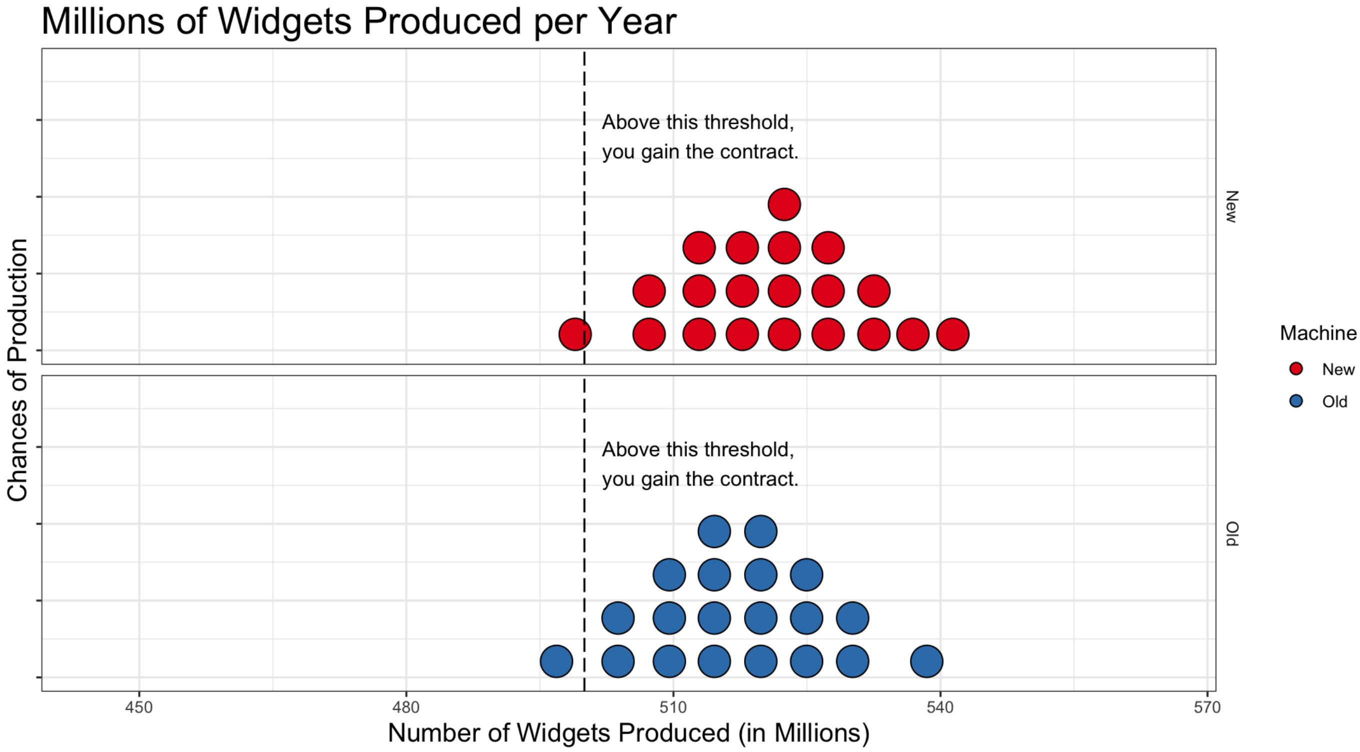
Quantile Dotplots

A chart function for quantile dotplots of the possible output for each machine.

```
quantile_dotplots <- function(df, data_domain, title, x_label, caption, decision_thre
shold, threshold_label) {
  plt <- df %>% select(-one_of(c("draws", "draw_n"))) %>% unnest() %>%
      ggplot(aes(x = quantiles, fill = Machine)) +
      geom_dotplot(binwidth = 4, binaxis = "x", dotsize = .9, stackratio = 1.35) +
      theme_bw() +
      scale_fill_brewer(palette = "Set1") +
      facet_grid(Machine ~ .) +
      xlim(data_domain[1], data_domain[2]) +
      ylim(0, .075) +
      labs(
        title = title,
        x = x_label,
        y = "Chances of Production",
        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)) +
      geom_vline(xintercept = decision_threshold, linetype = "longdash") + # contract
threshold
      annotate("text", x = (decision_threshold + 2), y = 0.05, label = threshold_labe
l, hjust = 0, vjust = 0)

  return(plt)
}

quantile_dotplots(df = gain_prac_df,
                  data_domain = data_domain_gain,
                  title = "Millions of Widgets Produced per Year",
                  x_label = "Number of Widgets Produced (in Millions)",
                  caption = "Each dot represents a 5% chance that different numbers o
f widgets (in millions) could be produced by each machine a given year.",
                  decision_threshold = 500,
                  threshold_label = "Above this threshold,\nyou gain the contract.")
```



Each dot represents a 5% chance that different numbers of widgets (in millions) could be produced by each machine a given year.

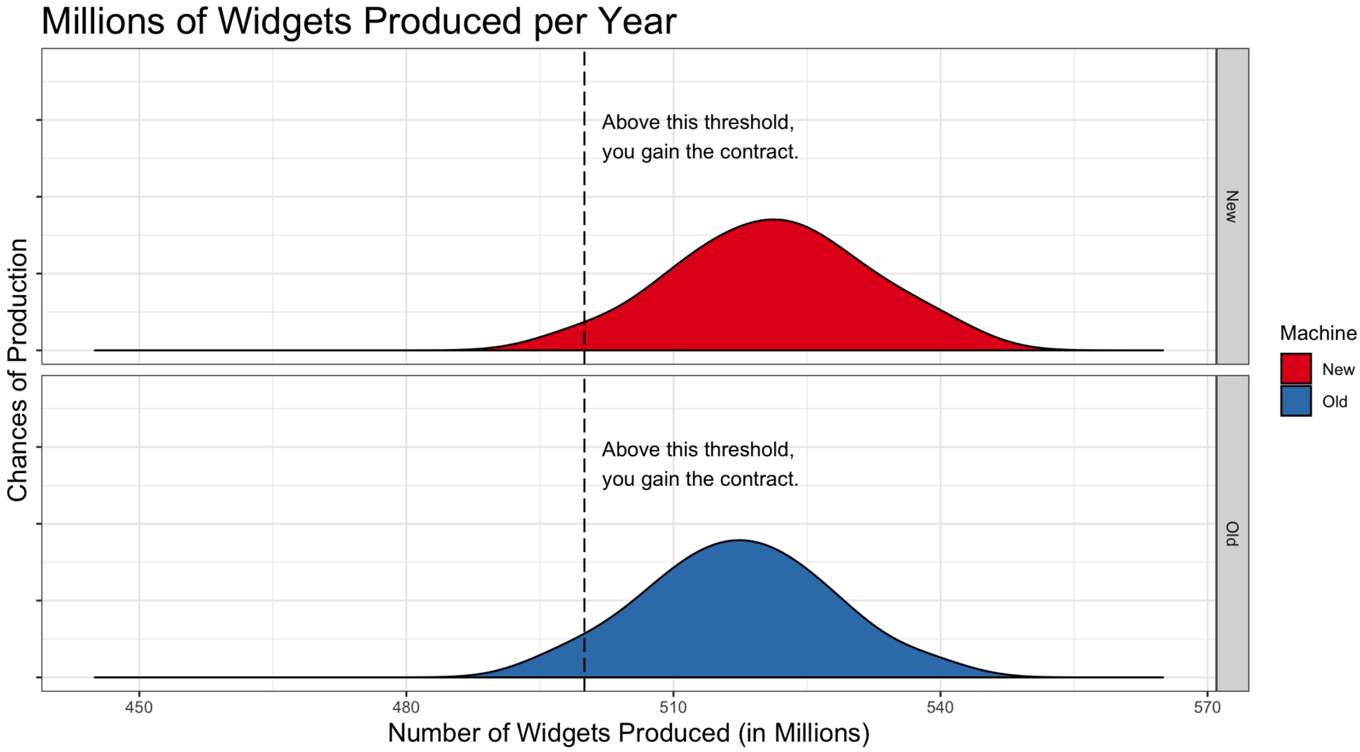
Densities

A chart function for continuous probability densities of the possible output for each machine.

```
densities <- function(df, data_domain, title, x_label, caption, decision_threshold, t
hreshold_label) {
  plt <- df %>% select(-one_of(c("draws", "draw_n"))) %>% unnest() %>%
      ggplot(aes(x = quantiles, fill = Machine)) +
      geom_density() +
      theme_bw() +
      scale_fill_brewer(palette = "Set1") +
      facet_grid(Machine ~ .) +
      xlim(data_domain[1], data_domain[2]) +
      ylim(0, .075) +
      labs(
        title = title,
        x = x_label,
        y = "Chances of Production",
        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)) +
      geom_vline(xintercept = decision_threshold, linetype = "longdash") + # contract
threshold
      annotate("text", x = (decision_threshold + 2), y = 0.05, label = threshold_labe
l, hjust = 0, vjust = 0)

  return(plt)
}

densities(df = gain_prac_df,
          data_domain = data_domain_gain,
          title = "Millions of Widgets Produced per Year",
          x_label = "Number of Widgets Produced (in Millions)",
          caption = "The height of the shape represents the chances that different nu
mbers of widgets (in millions) could be produced by each machine a given year.",
          decision_threshold = 500,
          threshold_label = "Above this threshold,\nyou gain the contract.")
```



The height of the shape represents the chances that different numbers of widgets (in millions) could be produced by each machine a given year.

Stimuli Generation

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

```
# 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

# 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)

  if (all(use_df$frame=="gain")) { # stimuli for gain framing trials
    # means only
    plt <- means_only(df = use_df,
      data_domain = data_domain_gain,
      title = "Millions of Widgets Produced per Year",
      x_label = "Average Number of Widgets Produced (in Millions)",
      caption = "Dots represent the average number of widgets (in millions) that coul
d be produced by each machine in a given year.",
      decision_threshold = 500,
      threshold_label = "Above this threshold,\nyou gain the contract.")
    fname <- paste("stimuli/means_only-", c,".svg", sep = "")
    ggsave(file = fname, plot = plt, width = dims[1], height = dims[2])

    # intervals only
    plt <- intervals_only(df = use_df,
      data_domain = data_domain_gain,
      title = "Millions of Widgets Produced per Year",
      x_label = "Range of Widgets Produced (in Millions)",
      caption = "Intervals contain 95% of the possible numbers of widgets (in million
s) that could be produced by each machine in a given year.",
      decision_threshold = 500,
      threshold_label = "Above this threshold,\nyou gain the contract.")
    fname <- paste("stimuli/intervals_only-", 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_gain,
      title = "Millions of Widgets Produced per Year",
      x_label = "Range of Widgets Produced (in Millions)",
      caption = "Dots represent the average number of widgets (in millions) that coul
d be produced by each machine in a given year. Intervals contain 95% of the possible
 numbers of widgets that could be produced in a given year.",
      decision_threshold = 500,
      threshold_label = "Above this threshold,\nyou gain the contract.")
    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_gain,
      title = "Millions of Widgets Produced per Year",
      x_label = "Number of Widgets Produced (in Millions)",
      caption = "Lines represent the number of widgets (in millions) that could be pr
oduced by each machine in a given year.",
```

```
      decision_threshold = 500,
      threshold_label = "Above this threshold,\nyou gain the contract.",
      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_gain,
      title = "Millions of Widgets Produced per Year",
      x_label = "Number of Widgets Produced (in Millions)",
      caption = "Dots represent the average number of widgets (in millions) that coul
d be produced by each machine in a given year. Lines represent the number of widgets
 that could produced in a given year.",
      decision_threshold = 500,
      threshold_label = "Above this threshold,\nyou gain the contract.",
      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,
      data_domain = data_domain_gain,
      title = "Millions of Widgets Produced per Year",
      x_label = "Number of Widgets Produced (in Millions)",
      caption = "Each dot represents a 5% chance that different numbers of widgets (i
n millions) could be produced by each machine a given year.",
      decision_threshold = 500,
      threshold_label = "Above this threshold,\nyou gain the contract.")
    fname <- paste("stimuli/QDPs-", c, ".svg", sep = "")
    ggsave(file = fname, plot = plt, width = dims[1], height = dims[2])

    # densities
    plt <- densities(df = use_df,
      data_domain = data_domain_gain,
      title = "Millions of Widgets Produced per Year",
      x_label = "Number of Widgets Produced (in Millions)",
      caption = "The height of the shape represents the chances that different number
s of widgets (in millions) could be produced by each machine a given year.",
      decision_threshold = 500,
      threshold_label = "Above this threshold,\nyou gain the contract.")
    fname <- paste("stimuli/densities-", c, ".svg", sep = "")
    ggsave(file = fname, plot = plt, width = dims[1], height = dims[2])


  } else { # stimuli for loss framing trials
    # means only
    plt <- means_only(df = use_df,
      data_domain = data_domain_loss,
      title = "Defective Widgets per Million Produced per Year",
      x_label = "Average Number of Defective Widgets per Million Produced",
      caption = "Dots represent the average number of defective widgets per million t
hat could be produced by each machine in a given year.",
      decision_threshold = 75,
      threshold_label = "Above this threshold,\nyou lose the contract.")
    fname <- paste("stimuli/means_only-", c,".svg", sep = "")
```

```
    ggsave(file = fname, plot = plt, width = dims[1], height = dims[2])

    # intervals only
    plt <- intervals_only(df = use_df,
      data_domain = data_domain_loss,
      title = "Defective Widgets per Million Produced per Year",
      x_label = "Range of Defective Widgets per Million Produced",
      caption = "Intervals contain 95% of the possible numbers of defective widgets p
er million that could be produced by each machine in a given year.",
      decision_threshold = 75,
      threshold_label = "Above this threshold,\nyou lose the contract.")
    fname <- paste("stimuli/intervals_only-", 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_loss,
      title = "Defective Widgets per Million Produced per Year",
      x_label = "Number of Defective Widgets per Million Produced",
      caption = "Dots represent the average number of defective widgets per million t
hat could be produced by each machine in a given year. Intervals contain 95% of the p
ossible numbers of defective widgets per million that could be produced in a given ye
ar.",
      decision_threshold = 75,
      threshold_label = "Above this threshold,\nyou lose the contract.")
    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_loss,
      title = "Defective Widgets per Million Produced per Year",
      x_label = "Number of Defective Widgets per Million Produced",
      caption = "Lines represent the number of defective widgets per million that cou
ld be produced by each machine in a given year.",
      decision_threshold = 75,
      threshold_label = "Above this threshold,\nyou lose the contract.",
      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_loss,
      title = "Defective Widgets per Million Produced per Year",
      x_label = "Number of Defective Widgets per Million Produced",
      caption = "Dots represent the average number of defective widgets per million t
hat could be produced by each machine in a given year. Lines represent number of defe
ctive widgets per million that could be produced in a given year.",
      decision_threshold = 75,
      threshold_label = "Above this threshold,\nyou lose the contract.",
      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,
      data_domain = data_domain_loss,
      title = "Defective Widgets per Million Produced per Year",
      x_label = "Number of Defective Widgets per Million Produced",
      caption = "Each dot represents a 5% chance that different numbers of defective
 widgets per million could be produced by each machine a given year.",
      decision_threshold = 75,
      threshold_label = "Above this threshold,\nyou lose the contract.")
    fname <- paste("stimuli/QDPs-", c, ".svg", sep = "")
    ggsave(file = fname, plot = plt, width = dims[1], height = dims[2])

    # densities
    plt <- densities(df = use_df,
      data_domain = data_domain_loss,
      title = "Defective Widgets per Million Produced per Year",
      x_label = "Number of Defective Widgets per Million Produced",
      caption = "The height of the shape represents the chances that different number
s of defective widgets per million could be produced by each machine a given year.",
      decision_threshold = 75,
      threshold_label = "Above this threshold,\nyou lose the contract.")
    fname <- paste("stimuli/densities-", c, ".svg", sep = "")
    ggsave(file = fname, plot = plt, width = dims[1], height = dims[2])
  }
}
```