Replicating Cleveland and McGill's study on Prolific - Data Analysis

Team Prolific (Aaron, Ethan, Hung, Noëlle)

October 03, 2019

Analysis of data collected from Cleveland and McGill replication experiment run on Prolific

References:
- https://github.com/mjskay/tidybayes
- @codementum

Libraries needed

```
library("jsonlite")
library(RCurl)
library(plyr)
library(tidyverse)
```

Grab the JSON file from firebase and convert into a list in R

```
tables <- read_json("https://clevelandmcgill-c116c.firebaseio.com/.json", simplifyVector = TRUE)
# str(tables)
# tables$Session
```

Create the sessions tibble

```
sessions <- tibble()
for (session in tables$Session) {
  new_row <- as_tibble(session)
  sessions <- sessions %>% bind_rows(new_row)
}

# Filter out empty response
sessions <- sessions %>%
  filter(age != "" & animal != "" & country != "" & prolific_id != "" & session_id != "")

# sessions
```

Create the trials tibble

```r
trials = tibble()
for (img_name in tables$Trials) {
  for (observation in img_name) {
      new_row <- as_tibble(observation)
      trials <- trials %>% bind_rows(new_row)
    # trials <- rbind(trials_df, data.frame(observation))
  }
}

# Filter out empty responses
trials <- trials %>%
  filter(actual_answer != "")

# trials
```

Clean the data to make later function calls easier

Create `condition` column from `image_name` (to easily filter by condition)

```r
trials$condition <- NA
for (i in seq_along(trials$image_name)) {
  if (substr(trials$image_name[i], 3, 3) == 'B')
    condition <- 'Bar'
  else if (substr(trials$image_name[i], 3, 3) == 'P')
    condition <- 'Pie'
  else
    condition <- '???'

  trials$condition[i] <- condition
}
```

Create column for the log__2 error rate (as described in the paper by Cleveland and McGill)

```r
trials$log2_error <- log(abs(strtoi(trials$actual_answer) - strtoi(trials$expected_answer)) + 0.125, ba
```

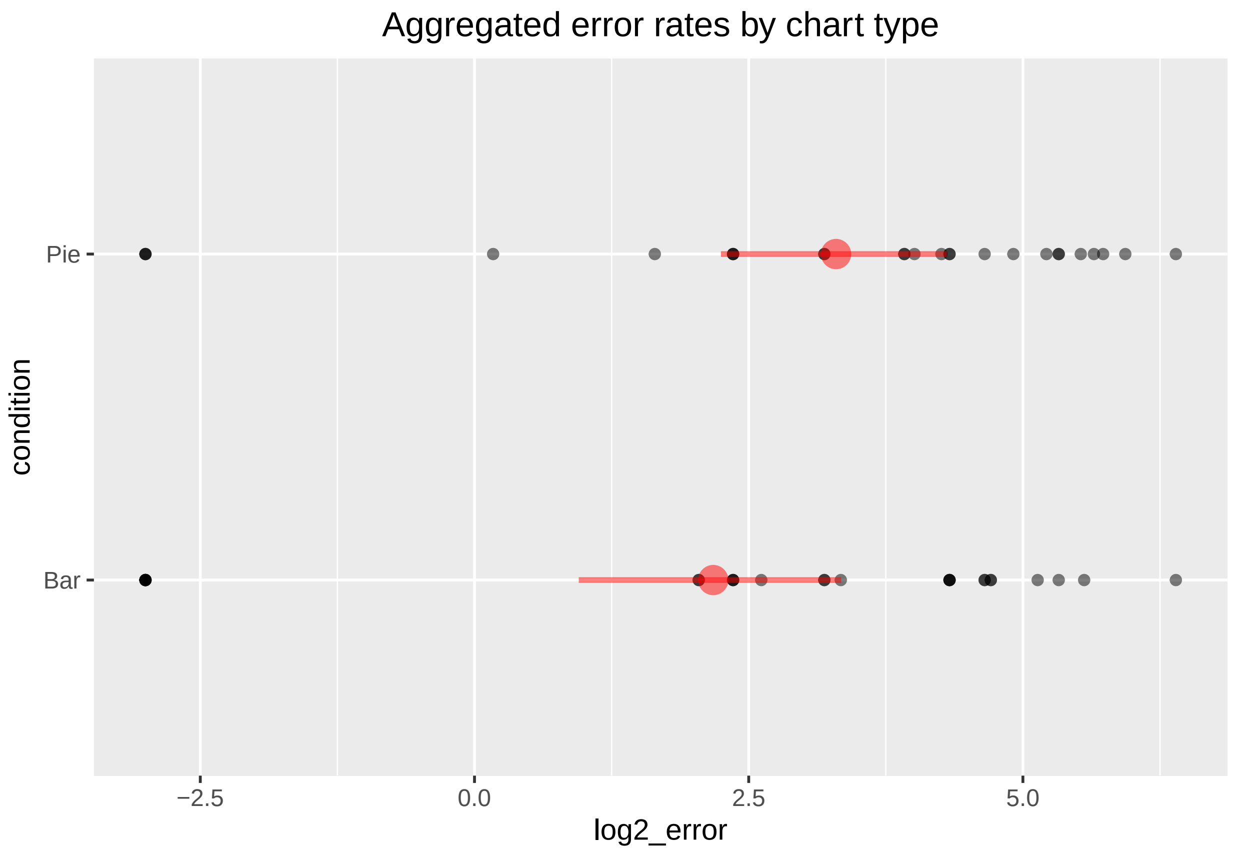
Create `participant` column from `session_id` (to make the facet plot tidier)

```r
trials$participant <- factor(
  trials$session_id, levels=unique(trials$session_id), labels = seq_along(unique(trials$session_id))
)

# trials
```

Comparing aggregate error rates for bar and pie charts

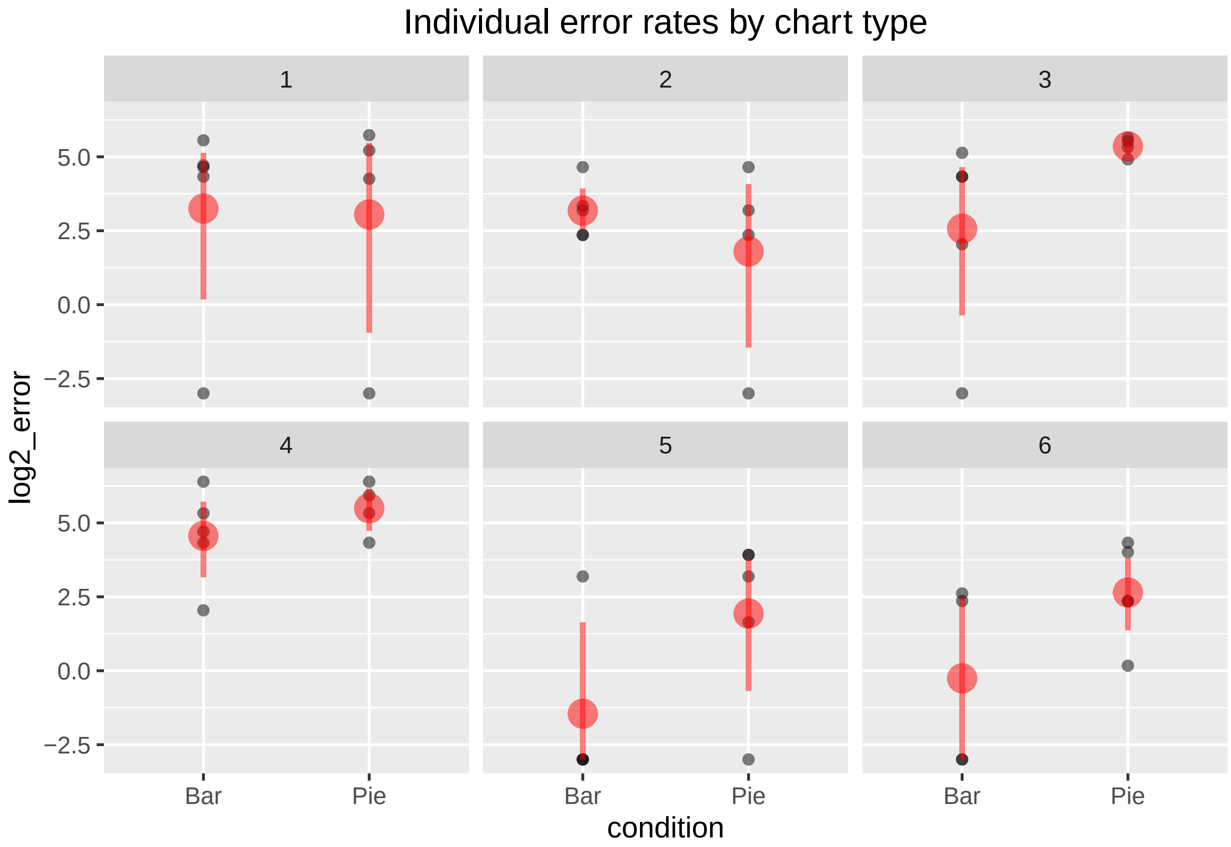
```
trials %>%
  ggplot(aes(x = condition, y = log2_error)) +
  geom_point(alpha = 0.5) +
  stat_summary(fun.data = "mean_cl_boot", colour = "red", size = 1.0, alpha=0.5) +
  coord_flip() +
  theme(plot.title = element_text(hjust = 0.5)) +
  ggtitle("Aggregated error rates by chart type")
```



As expected, error rates on pie charts are higher than the error rates on bar charts

Comparing individual participant error rates by chart type

```
trials %>%
  ggplot(aes(x = condition, y = log2_error)) +
  geom_point(alpha = 0.5) +
  stat_summary(fun.data = "mean_cl_boot", colour = "red", size = 1.0, alpha=0.5) +
  facet_wrap(~ participant) +
  theme(plot.title = element_text(hjust = 0.5)) +
  ggtitle("Individual error rates by chart type")
```



```
  # theme(panel.spacing = unit(2, "lines"))

ggsave('barvpie_byparticipant.pdf', units="in", width=5, height=8)
```

We still see the same trend of higher error rates on pie charts (with a couple of outliers)

Statistical Analysis

Normality test of the bar and pie responses

```
pie <- trials %>%
  filter(condition == "Pie")
bar <- trials %>%
  filter(condition == "Bar")

shapiro.test(bar$log2_error)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  bar$log2_error
## W = 0.80719, p-value = 0.0001433
```

```
shapiro.test(pie$log2_error)
```

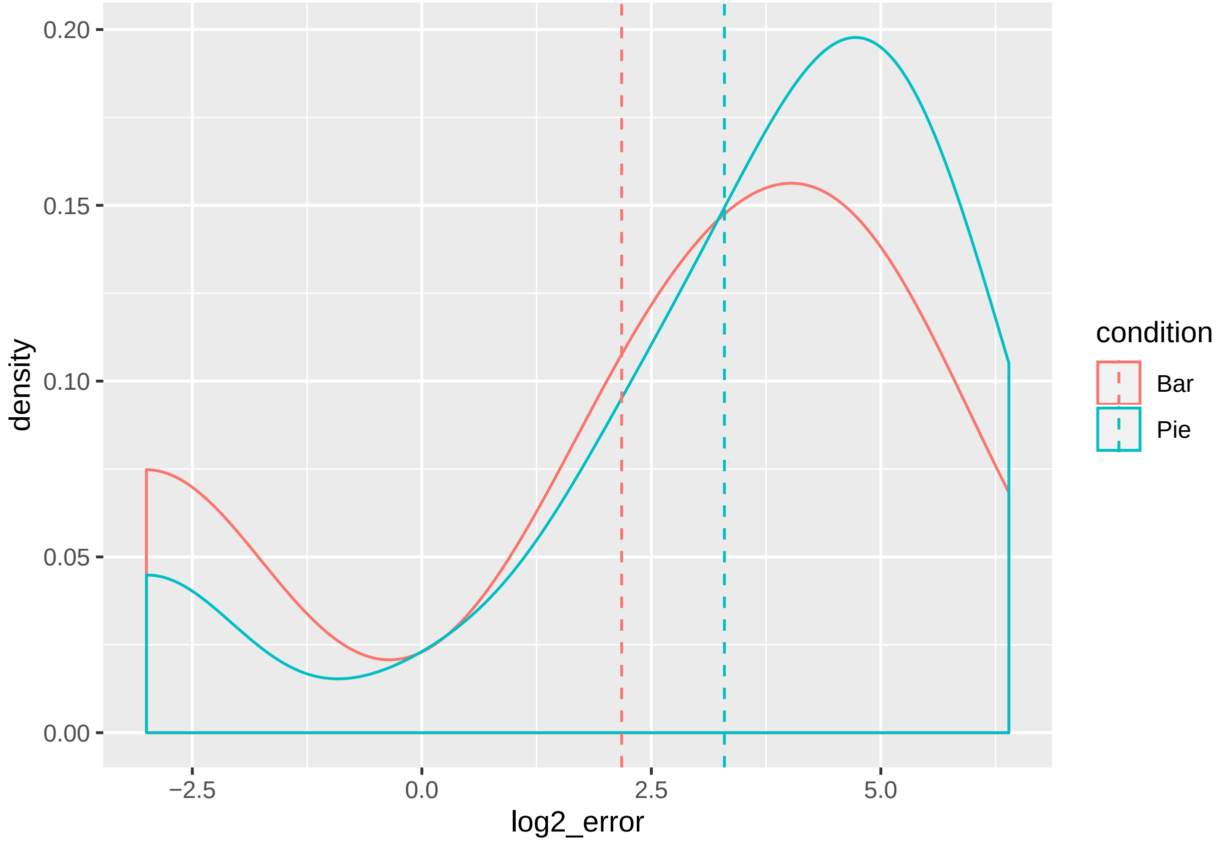
```
##
##  Shapiro-Wilk normality test
##
## data:  pie$log2_error
## W = 0.82182, p-value = 0.0004237
```

Density plots - validating normality test

Our normality test indicates that the error rates for the bar and pie charts are not normally distributed. We validate this by creating density plots by chart type, with the mean value of each group also indicated.

```
mu <- ddply(trials, "condition", summarise, grp.mean=mean(log2_error))

trials %>%
  ggplot(aes(x=log2_error, color=condition)) +
  geom_density() +
  geom_vline(data=mu, aes(xintercept=grp.mean, color=condition),
             linetype="dashed")
```



Wilcoxon rank sum test

Our test for normality came out negative therefore we perform the Wilcoxon rank sum test (Note that we might have had a normal distribution if we got rid of the outliers from participant 1)

```
wilcox.test(pie$log2_error, bar$log2_error)
```

```
## Warning in wilcox.test.default(pie$log2_error, bar$log2_error): cannot
## compute exact p-value with ties
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  pie$log2_error and bar$log2_error
## W = 439.5, p-value = 0.192
## alternative hypothesis: true location shift is not equal to 0
```

```
# t.test(pie$response, bar$response)
```