

Kasdin et al. (2025) show that dopamine in the brains of young zebra finches acts as a learning signal, increasing when they sing closer to their adult song and decreasing when they sing further away, effectively guiding their vocal development through trial-and-error. This suggests that complex natural behaviors, like learning to sing, are shaped by dopamine-driven reinforcement learning, similar to how artificial intelligence learns. You can find the paper at this link: <https://www.nature.com/articles/s41586-025-08729-1>.

Note they measure dopamine using fibre photometry, changes in the fluorescence indicate dopamine changes in realtime. Their specific measurement considers changes in fluorescence in 100-ms windows between 200 and 300 ms from the start of singing, averaged across development.

1. Using the `pwr` package for R (Champely, 2020), conduct a power analysis. How many observations would the researchers need to detect a moderate-to-large effect ($d = 0.65$) when using $\alpha = 0.05$ and default power (0.80) for a two-sided one sample t test.

***Response:** `n` should be 20/21.*

```
# Perform a power analysis for a one-sample t-test with a medium effect size (d = 0.65)
# Desired power = 0.80, two-sided alternative hypothesis
pwr.t.test(d = 0.65,
           power = 0.80,
           type = "one.sample",
           alternative = "two.sided")

##
##      One-sample t test power calculation
##
##              n = 20.58039
##              d = 0.65
##      sig.level = 0.05
##              power = 0.8
##      alternative = two.sided
```

2. Click the link to go to the paper. Find the source data for Figure 2. Download the Excel file. Describe what you needed to do to collect the data for Figure 2(g). Note that you only need the `closer_vals` and `further_vals`. Ensure to `mutate()` the data to get a difference (e.g., `closer_vals - further_vals`).

```
# Load data from CSV files for "Closer" and "Farther" conditions
Closer.dat <- read_csv("C_vals.csv")
Farther.dat <- read_csv("F_vals.csv")

# Combine the data into a single tibble and calculate the difference between Closer and Farther values
full.dat <- tibble(
  Farther = Farther.dat$data,
  Closer = Closer.dat$data
) %>%
  mutate(
    Difference = Closer - Farther # Calculate the difference
  )
```

3. Summarize the data.

- (a) Summarize the further data. Do the data suggest that dopamine in the brains of young zebra finches decreases when they sing further away?

***Response:** The data below suggests that the dopamine in the brains of young zebra finches does in fact decrease when they sing further away.*

- (b) Summarize the closer data. Do the data suggest that dopamine in the brains of young zebra finches increases when they sing closer to their adult song?

***Response:** The data below suggests that the dopamine in the brains of young zebra finches does in fact increase when they sing closer.*

- (c) Summarize the paired differences. Do the data suggest that there is a difference between dopamine in the brains of young zebra finches when they sing further away compared to closer to their adult song?

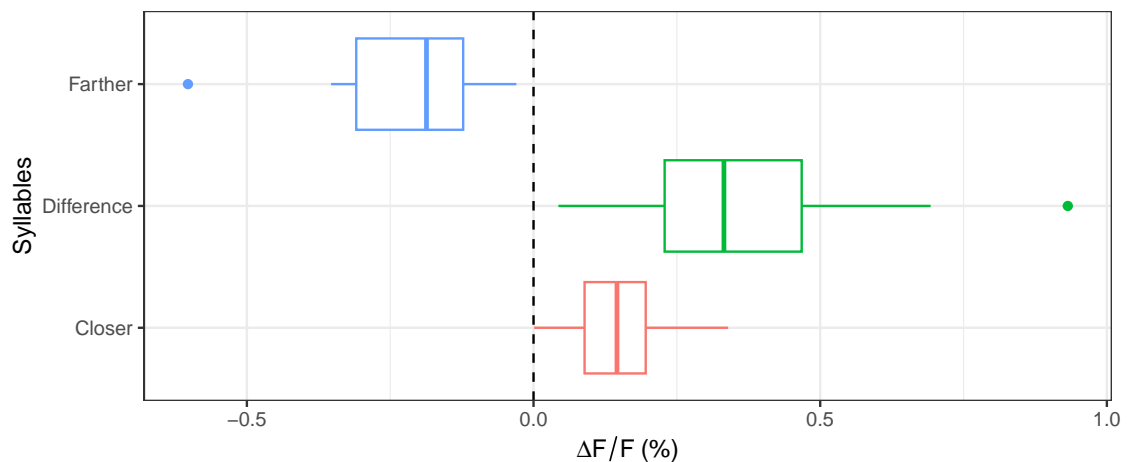
***Response:** The data below suggests that there exists a difference in the brains of young zebra finches singing closer in contrast to singing further away, as the plot

is more skewed towards greater dopamine when singing closer than lower dopamine when singing farther.*

Solution to a, b, and c:

```
# Reshape data to a long format for plotting
plot.full.dat <- pivot_longer(
  full.dat,
  cols = everything(),      # Reshape all columns to long format
  names_to = "type",        # Column names become the 'type' variable
  values_to = "values"      # Column values become the 'values' variable
)

# Create a boxplot of the 'Closer' and 'Farther' values
ggplot(data = plot.full.dat) +
  geom_boxplot(aes(x = type, y = values, color = type), show.legend = FALSE) + # Boxplot with color by 'type'
  coord_flip() + # Flip axes for horizontal boxplot
  theme_bw() + # Use black-and-white theme
  labs(x = "Syllables", # Label the x-axis
       y = expression(Delta*F/F~"(%)" ) + # Label the y-axis with a mathematical expression
  ) +
  geom_hline(yintercept = 0, linetype = "dashed") # Add a dashed horizontal line at y = 0
```

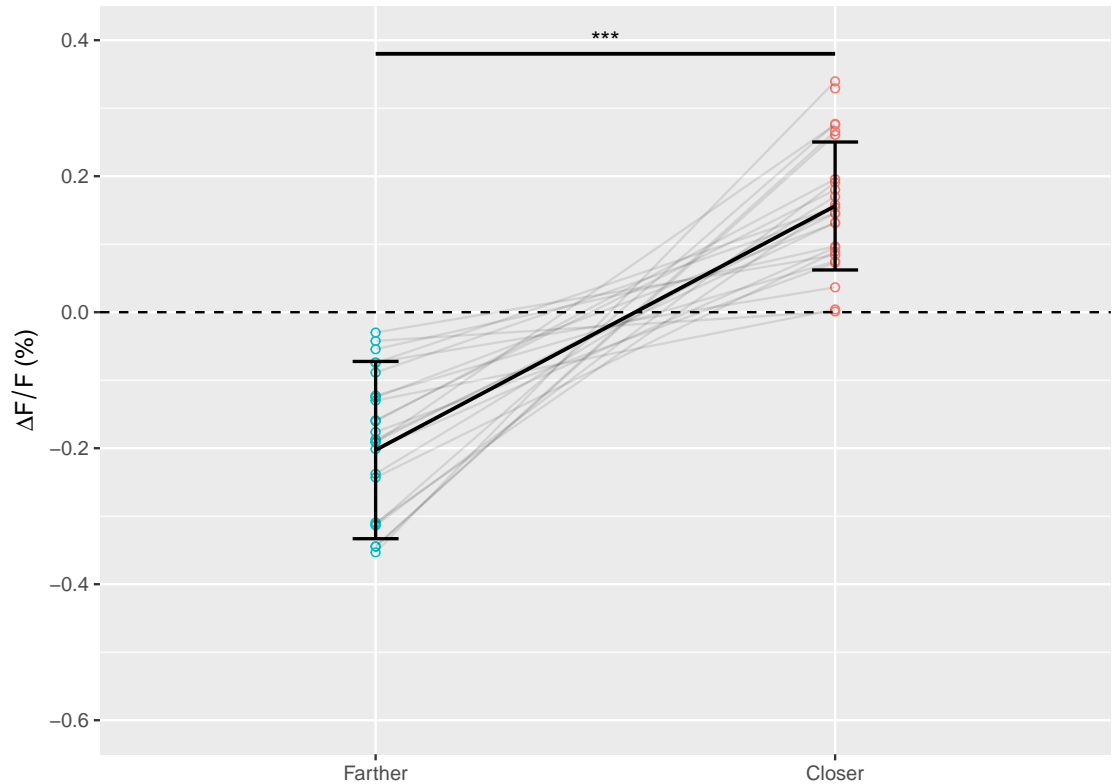


- (d) **Optional Challenge:** Can you reproduce Figure 2(g)? Note that the you can use `geom_errorbar()` to plot the range created by adding the mean \pm one standard deviation.

```
# Calculate mean and standard deviation for 'Closer' and 'Farther', then filter the data
plot.full.dat.error <- plot.full.dat %>%
  group_by(type) %>%
  summarize(mean_value = mean(values), # Mean of values
            sd_value = sd(values)) %>% # Standard deviation of values
  filter(type %in% c("Closer", "Farther")) # Filter for Closer and Farther types

# Modify the data to add a 'pair_group' variable for pairing data points
plot.full.dat <- plot.full.dat %>%
  filter(type %in% c("Closer", "Farther")) %>%
  mutate(pair_group = rep(1:25, each = 2)) # Create a group for pairs of data

# Create a scatterplot with error bars and mean line for 'Closer' and 'Farther'
ggplot(data = plot.full.dat) +
  geom_point(aes(x = type, y = values, color = type), shape = 1, show.legend = FALSE) + # Scatter points
  geom_line(aes(x = type, y = values, group = pair_group), alpha = 0.1) + # Lines connecting paired points
  geom_errorbar(data = plot.full.dat.error, aes(
    x = type,
    ymin = mean_value - sd_value, # Lower error bound
    ymax = mean_value + sd_value # Upper error bound
  ), width = 0.1,
  size = .7) + # Error bars
  geom_line(data = plot.full.dat.error, aes(x = type, y = mean_value, group = 1), size = .8) + # Mean line
  geom_hline(yintercept = 0, linetype = "dashed") + # Horizontal line at y = 0
  labs(x = NULL, # Remove x-axis label
       y = expression(Delta*F/F~"(%)" ) + # Y-axis label
  ) +
  ylim(c(-0.6, 0.4)) + # Set y-axis limits
  scale_x_discrete(limits = c("Farther", "Closer")) + # Set x-axis categories
  annotate("segment", x = "Farther", y = 0.38, xend = "Closer", yend = 0.38, size = .8) + # Line annotation
  annotate("text", x = 1.5, y = .4, label = "***", size = 4) # Text annotation
```



4. Conduct the inferences they do in the paper. Make sure to report the results a little more comprehensively – that is your parenthetical should look something like: ($t = 23.99$, $p < 0.0001$; $g = 1.34$; 95% CI: 4.43, 4.60).

Note: Your numbers may vary slightly as they performed some unclear correction of their p -values. I'm waiting to hear back from them via email!

- (a) “The close responses differed significantly from 0 ($p = 1.63 \times 10^{-8}$).”

***Response:** ($t = 8.30$; $p < 0.0001$; $g = 1.61$; 95% CI: 0.11, 0.19)*

```
# Perform t-test for 'Closer' data, testing if the mean is 0 (for CI)
t.test(full.dat$Closer,
       mu = 0,
       alternative = 'two.sided')

# Perform t-test for 'Closer' data, testing if the mean is greater than 0
t.test(full.dat$Closer,
       mu = 0,
       alternative = 'greater')

# Calculate Hedges' g for the 'Closer' data
hedges_g(full.dat$Closer)
```

- (b) “The far responses differed significantly from 0 ($p = 5.17 \times 10^{-8}$).”

***Response:** ($t = -7.71$; $p < 0.0001$; $g = -1.51$; 95% CI: -0.25, -0.14)*

```
# Perform t-test for 'Farther' data, testing if the mean is 0 (for CI)
t.test(full.dat$Farther,
       mu = 0,
       alternative = 'two.sided')

# Perform t-test for 'Farther' data, testing if the mean is less than 0
t.test(full.dat$Farther,
       mu = 0,
       alternative = 'less')

# Calculate Hedges' g for the 'Farther' data
hedges_g(full.dat$Farther)
```

- (c) “The difference between populations was significant ($p = 1.04 \times 10^{-8}$).”
***Response:** ($t = 8.51$; $p < 0.0001$; $g = 1.65$; 95% CI: 0.27, 0.44)*

```
# Perform t-test for the 'Difference' data, testing if the mean is 0
t.test(full.dat$Difference,
       mu = 0,
       alternative = 'two.sided')

# Calculate Hedges' g for the 'Difference' data
hedges_g(full.dat$Difference)
```

5. Reverse engineer the hypothesis test plot from Lecture 20 to create accurate hypothesis testing plots for each part of the previous question.

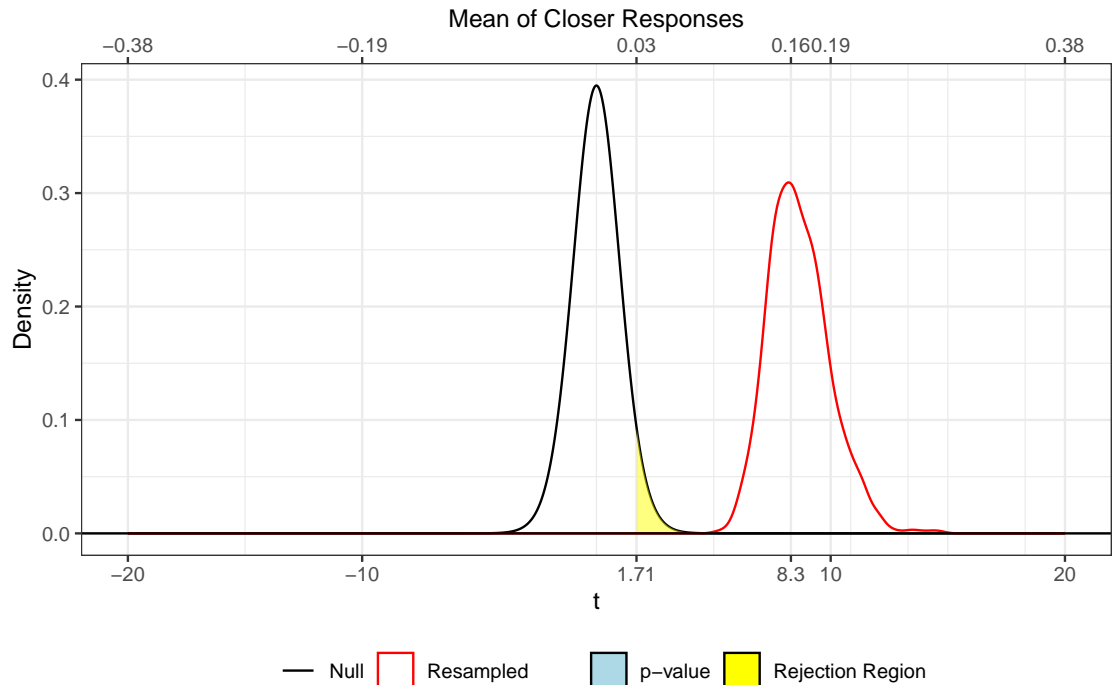
- (a) Question 4, part(a).

```
# Set t-statistic breaks and calculate corresponding x-bar breaks
t.breaks <- c(-20, -10, 10, 20,
              qt(1 - 0.05, df = 24),
              8.3024)
xbar.breaks <- t.breaks * sd(full.dat$Closer) / sqrt(length(full.dat$Closer))

# Create a plot with t-distribution, resampled t-statistics, and p-value areas for Closer
ggplot() +
  geom_line(data = t.dist.dat, aes(x = x, y = ts, color = "Null")) +
  geom_ribbon(data = subset(t.dist.dat, x >= qt(1 - 0.05, df = 24)),
            aes(x = x, ymin = 0, ymax = ts, fill = "Rejection Region"),
            alpha = 0.5) +
  geom_ribbon(data = subset(t.dist.dat, x >= 8.3024),
            aes(x = x, ymin = 0, ymax = ts, fill = "p-value"),
            alpha = 0.25) +
  geom_density(data = resamples.Closer,
              aes(x = t.stat, color = "Resampled")) +
  geom_hline(yintercept = 0) +
  scale_color_manual("", values = c("black", "red")) +
  scale_fill_manual("", values = c("lightblue", "yellow")) +
  theme_bw() +
  theme(legend.position = "bottom") +
  ggtitle("T-Test for the Closer Responses Population Mean",
         subtitle = bquote(H[0]:mu[X] == 0 ~ "vs" ~ H[a]:mu[X] > 0)) +
  scale_x_continuous("t",
                    breaks = t.breaks, labels = round(t.breaks, 2),
                    sec.axis = sec_axis(~ .,
                                       breaks = t.breaks, labels = round(xbar.breaks, 2),
                                       name = "Mean of Closer Responses")) +
  labs(x = "t",
       y = "Density")
```

T-Test for the Closer Responses Population Mean

$H_0: \mu_X = 0$ vs $H_a: \mu_X > 0$



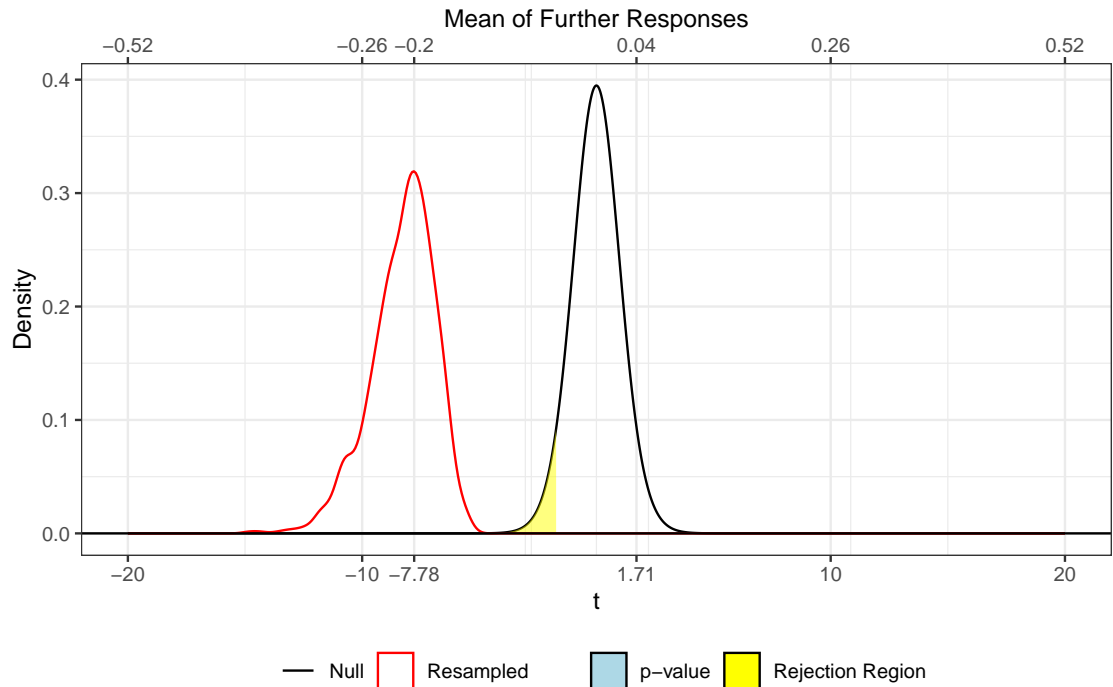
(b) Question 4, part(b).

```
# Set t-statistic breaks and calculate corresponding x-bar breaks for Farther
t.breaks <- c(-20, -10, 10, 20,
             qt(1 - 0.05, df = 24),
             -7.779)
xbar.breaks <- t.breaks * sd(full.dat$Farther) / sqrt(length(full.dat$Farther))

# Create a plot with t-distribution, resampled t-statistics, and p-value areas for Farther
ggplot() +
  geom_line(data = t.dist.dat, aes(x = x, y = ts, color = "Null")) +
  geom_ribbon(data = subset(t.dist.dat, x <= qt(0.05, df = 24)),
            aes(x = x, ymin = 0, ymax = ts, fill = "Rejection Region"),
            alpha = 0.5) +
  geom_ribbon(data = subset(t.dist.dat, x >= 8.3024),
            aes(x = x, ymin = 0, ymax = ts, fill = "p-value"),
            alpha = 0.25) +
  geom_density(data = resamples.Farther,
              aes(x = t.stat, color = "Resampled")) +
  geom_hline(yintercept = 0) +
  scale_color_manual("", values = c("black", "red")) +
  scale_fill_manual("", values = c("lightblue", "yellow")) +
  theme_bw() +
  theme(legend.position = "bottom") +
  ggtitle("T-Test for the Further Responses Population Mean",
         subtitle = bquote(H[0]:mu[X] == 0 ~ "vs" ~ H[a]:mu[X] < 0)) +
  scale_x_continuous("t",
                    breaks = t.breaks, labels = round(t.breaks, 2),
                    sec.axis = sec_axis(~ .,
                                       breaks = t.breaks, labels = round(xbar.breaks, 2),
                                       name = "Mean of Further Responses")) +
  labs(x = "t",
       y = "Density")
```

T-Test for the Further Responses Population Mean

$H_0: \mu_X = 0$ vs $H_a: \mu_X < 0$



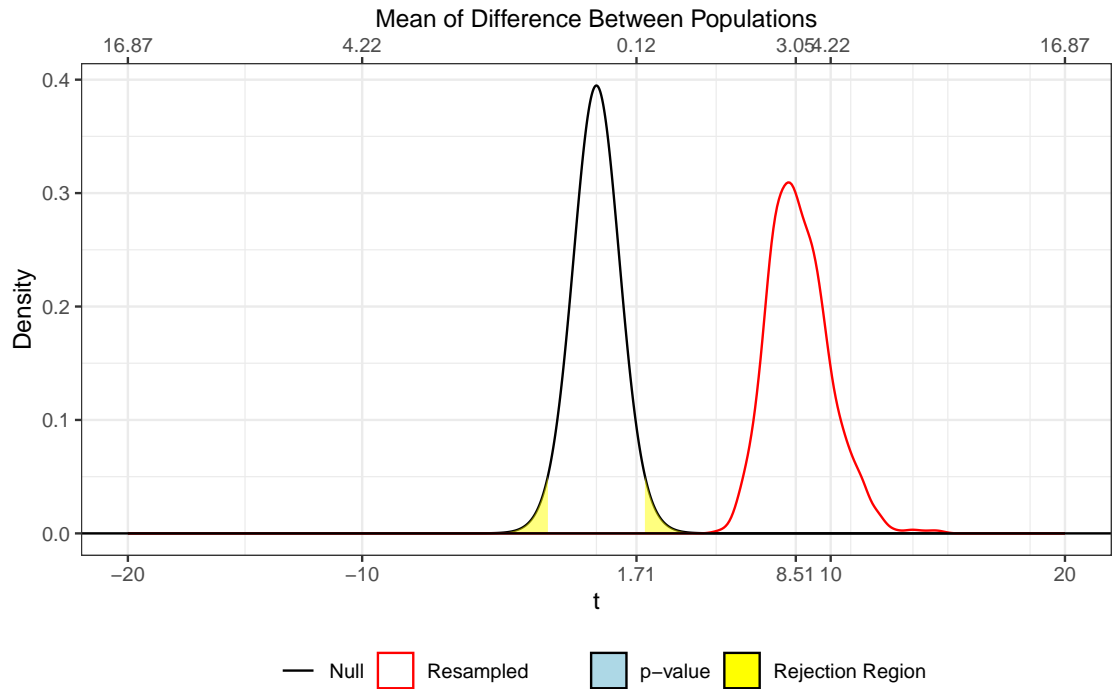
(c) Question 4, part(c).

```
# Set t-statistic breaks and calculate corresponding x-bar breaks for Difference
t.breaks <- c(-20, -10, 10, 20,
             qt(1 - 0.05, df = 24),
             8.5109)
xbar.breaks <- t.breaks * sd(full.dat$Difference) / sqrt(length(full.dat$Difference))

# Create a plot with t-distribution, resampled t-statistics, and p-value areas for Difference
ggplot() +
  geom_line(data = t.dist.dat, aes(x = x, y = ts, color = "Null")) +
  geom_ribbon(data = subset(t.dist.dat, x >= qt(1 - 0.025, df = 24)),
            aes(x = x, ymin = 0, ymax = ts, fill = "Rejection Region"),
            alpha = 0.5) +
  geom_ribbon(data = subset(t.dist.dat, x <= qt(0.025, df = 24)),
            aes(x = x, ymin = 0, ymax = ts, fill = "Rejection Region"),
            alpha = 0.5) +
  geom_ribbon(data = subset(t.dist.dat, x >= 8.3024),
            aes(x = x, ymin = 0, ymax = ts, fill = "p-value"),
            alpha = 0.25) +
  geom_density(data = resamples.Closer,
              aes(x = t.stat, color = "Resampled")) +
  geom_hline(yintercept = 0) +
  scale_color_manual("", values = c("black", "red")) +
  scale_fill_manual("", values = c("lightblue", "yellow")) +
  theme_bw() +
  theme(legend.position = "bottom") +
  ggtitle("T-Test for the Difference Between 'Populations' Population Mean",
         subtitle = bquote(H[0]:mu[X] == 0 ~ "vs" ~ H[a]:mu[X] != 0)) +
  scale_x_continuous("t",
                    breaks = t.breaks, labels = round(t.breaks, 2),
                    sec.axis = sec_axis(~.,
                                       breaks = t.breaks, labels = round(xbar.breaks, 2),
                                       name = "Mean of Difference Between Populations")) +
  labs(x = "t",
       y = "Density")
```

T-Test for the Difference Between ``Populations`` Population Mean

$H_0: \mu_X = 0$ vs $H_a: \mu_X \neq 0$



References

Champely, S. (2020). *pwr: Basic Functions for Power Analysis*. R package version 1.3-0.

Kasdin, J., Duffy, A., Nadler, N., Raha, A., Fairhall, A. L., Stachenfeld, K. L., and Gadagkar, V. (2025). Natural behaviour is learned through dopamine-mediated reinforcement. *Nature*, pages 1–8.