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 flourescence 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?

```
library(pwr)
set.seed(5656)

(pwr.t.test(d = 0.65,
           power = 0.80,
           sig.level = 0.05,
           alternative = "two.sided",
           type = "one.sample"))


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

   Since $n = 20.58039$, at least 21 observations are needed to detect a moderate-to-large effect.

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`).

```
library(tidyverse)

farther <- read_csv("farther.csv",col_names = F)
closer <- read_csv("closer.csv",col_names = F)

fig.2g.data <- tibble(g.Farther_vals = farther[[1]],
                      g.Closer_vals = closer[[1]]) |>
  mutate(difference = g.Closer_vals - g.Farther_vals)
#view(fig.2g.data)
```

   To collect the data for Figure 2(g), the g.Farther_vals and g.Closer_vals columns were downloaded, converted to csv files, and merged into a tibble. Their difference was then calculated and mutated as a third column.

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?

```
library(e1071)

farther.summary <- fig.2g.data |>
  summarize(mean       = mean(g.Farther_vals),
            sd         = sd(g.Farther_vals),
            median     = median(g.Farther_vals),
            IQR        = IQR(g.Farther_vals),
            skewness   = skewness(g.Farther_vals),
            exkurtosis = kurtosis(g.Farther_vals))

(farther.summary)
```

```
## # A tibble: 1 x 6
##     mean    sd median   IQR skewness exkurtosis
##    <dbl> <dbl>  <dbl> <dbl>    <dbl>      <dbl>
## 1 -0.203 0.130 -0.187 0.187    -1.04       1.19
```

Since the mean is less than zero, the data suggest that on average, dopamine in the brains of young zebra finches decreases 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?

```
closer.summary <- fig.2g.data |>
  summarize(mean       = mean(g.Closer_vals),
            sd         = sd(g.Closer_vals),
            median     = median(g.Closer_vals),
            IQR        = IQR(g.Closer_vals),
            skewness   = skewness(g.Closer_vals),
            exkurtosis = kurtosis(g.Closer_vals))

(closer.summary)

## # A tibble: 1 x 6
##    mean     sd median   IQR skewness exkurtosis
##   <dbl>  <dbl>  <dbl> <dbl>    <dbl>      <dbl>
## 1 0.156 0.0941  0.146 0.107    0.295     -0.859
```

Since the mean is greater than zero, the data suggest that on average, dopamine in the brains of young zebra finches increases when they sing closer to their adult song.

(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?

```
difference.summary <- fig.2g.data |>
  summarize(mean       = mean(difference),
            sd         = sd(difference),
            median     = median(difference),
            IQR        = IQR(difference),
            skewness   = skewness(difference),
            exkurtosis = kurtosis(difference))

(difference.summary)

## # A tibble: 1 x 6
##    mean    sd median   IQR skewness exkurtosis
##   <dbl> <dbl>  <dbl> <dbl>    <dbl>      <dbl>
## 1 0.359 0.211  0.332 0.239    0.773      0.128
```

Since the mean is significantly far from zero, it appears that there is a distinct difference between dopamine in the brains of young zebra finches when they sing further away compared to closer to their adult song.

(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 ± one standard deviation.

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 ($p = 1.63 \times 10^{-8}$)."

```
library(effectsize)

# a) Closer

mu0 <- 0
x <- fig.2g.data$g.Closer_vals
xbar <- mean(x)
s <- sd(x)
n <- length(x)
(t.stat.a <- (xbar - mu0)/(s/sqrt(n)))
```

```
## [1] 8.302409

(p.val <- 2*pt(q=-abs(t.stat.a), df = n-1))

## [1] 1.626307e-08

t.test(x=x, mu = mu0, alternative = "greater")

##
##  One Sample t-test
##
## data:  x
## t = 8.3024, df = 24, p-value = 8.132e-09
## alternative hypothesis: true mean is greater than 0
## 95 percent confidence interval:
##  0.1240301       Inf
## sample estimates:
## mean of x
## 0.1562231

(hedges_g(x = x, mu = mu0, alternative = "greater"))

## Hedges' g |      95% CI
## -----------------------
## 1.61      | [1.10, Inf]
##
## - One-sided CIs: upper bound fixed at [Inf].

interpret_hedges_g(1.61)

## [1] "large"
## (Rules: cohen1988)

t.test(x=x, mu = mu0, alternative = "two.sided")

##
##  One Sample t-test
##
## data:  x
## t = 8.3024, df = 24, p-value = 1.626e-08
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.1173875 0.1950586
## sample estimates:
## mean of x
## 0.1562231
```

The close responses differed significantly from 0 ($t = 8.3024$, $p = 1.626 \times 10^{-08}$; $g = 1.61$; 95% CI: 0.1173875, 0.1950586).

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

```
# b) Farther

mu0 <- 0
x <- fig.2g.data$g.Farther_vals
xbar <- mean(x)
s <- sd(x)
n <- length(x)
(t.stat.b <- (xbar - mu0)/(s/sqrt(n)))

## [1] -7.777991

(p.val <- 2*pt(q=-abs(t.stat.b), df = n-1))

## [1] 5.174718e-08

t.test(x=x, mu = mu0, alternative = "less")

##
##  One Sample t-test
##
## data:  x
## t = -7.778, df = 24, p-value = 2.587e-08
## alternative hypothesis: true mean is less than 0
## 95 percent confidence interval:
##       -Inf -0.1581322
## sample estimates:
##  mean of x
## -0.2027244
```

```
(hedges_g(x = x, mu = mu0, alternative = "less"))

## Hedges' g |        95% CI
## -------------------------
## -1.51     | [-Inf, -1.02]
##
## - One-sided CIs: lower bound fixed at [-Inf].

interpret_hedges_g(-1.51)

## [1] "large"
## (Rules: cohen1988)

t.test(x=x, mu = mu0, alternative = "two.sided")

##
##  One Sample t-test
##
## data:  x
## t = -7.778, df = 24, p-value = 5.175e-08
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.2565176 -0.1489313
## sample estimates:
##  mean of x
## -0.2027244
```

The far responses differed significantly from 0 ($t = -7.778$, $p = 5.175 \times 10^{-08}$; $g = -1.51$; 95% CI: -0.2565176, -0.1489313).

(c) "The difference between populations was significant ($p = 1.04 \times 10^{-8}$)."

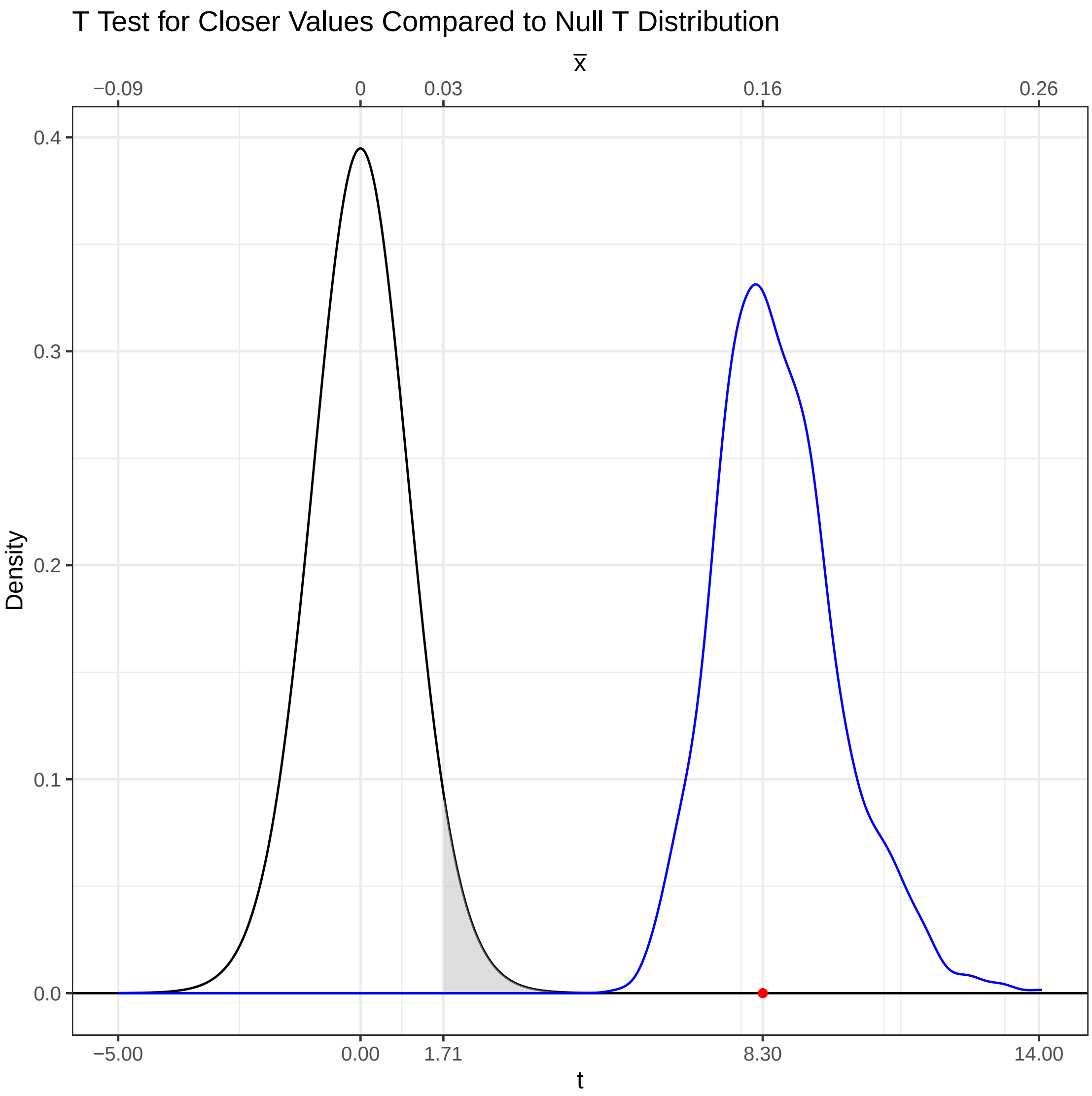
```
# c) Difference

mu0 <- 0
x <- fig.2g.data$difference
xbar <- mean(x)
s <- sd(x)
n <- length(x)
(t.stat.c <- (xbar - mu0)/(s/sqrt(n)))

## [1] 8.510932

(p.val <- 2*pt(q=-abs(t.stat.c), df = n-1))

## [1] 1.036907e-08

t.test(x=x, mu = mu0, alternative = "two.sided")

##
##  One Sample t-test
##
## data:  x
## t = 8.5109, df = 24, p-value = 1.037e-08
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.2719028 0.4459921
## sample estimates:
## mean of x
## 0.3589475

(hedges_g(x = x, mu = mu0, alternative = "two.sided"))

## Hedges' g |       95% CI
## ------------------------
## 1.65      | [1.04, 2.24]

interpret_hedges_g(1.65)

## [1] "large"
## (Rules: cohen1988)
```

The difference between populations was significant ($t = 8.5109$, $p = 1.037 \times 10^{-08}$; $g = 1.65$; 95% CI: 0.2719028 0.4459921).

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).

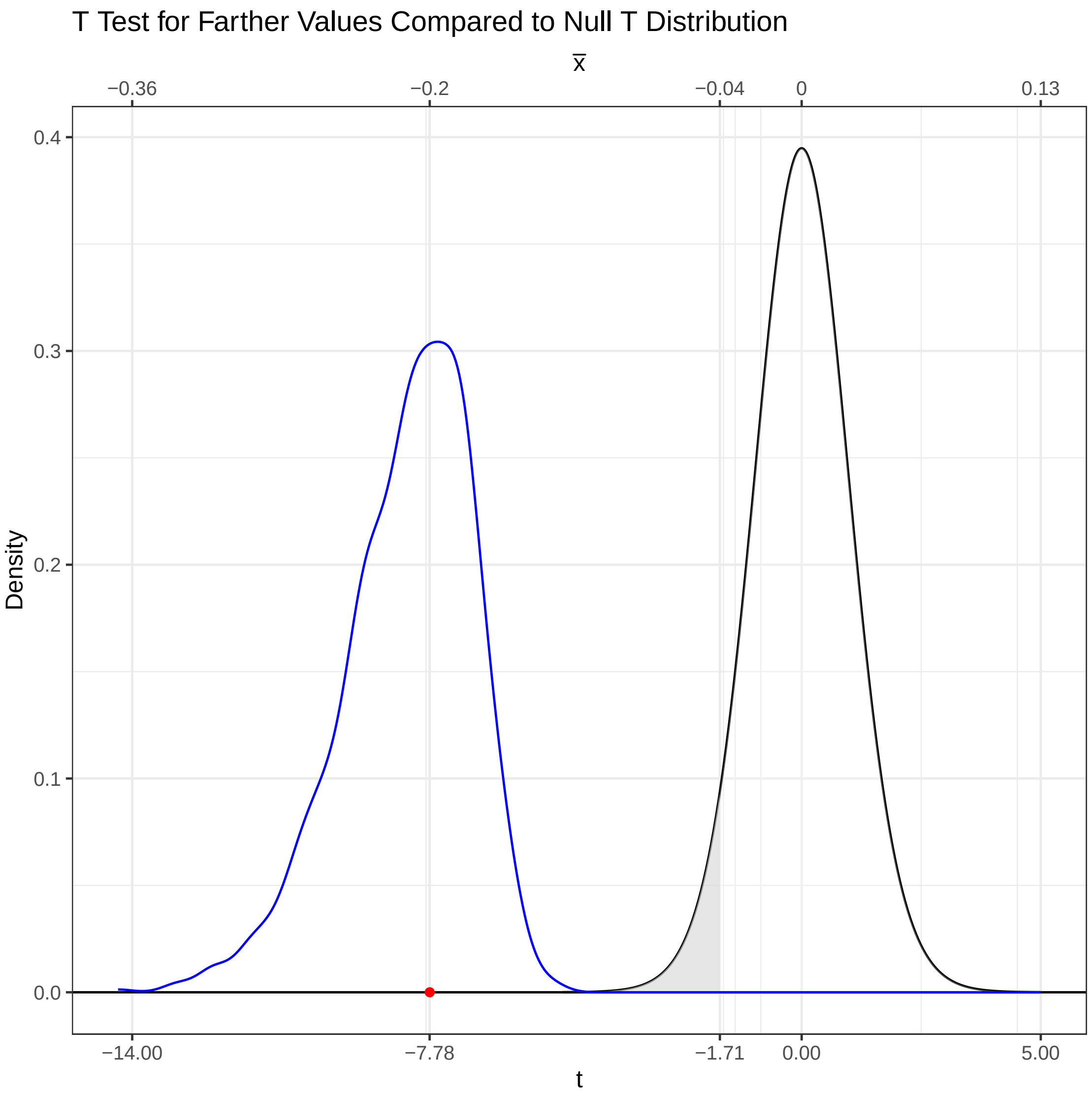
```r
# a) Closer

mu0 <- 0
x <- fig.2g.data$g.Closer_vals
xbar <- mean(x)
s <- sd(x)
n <- length(x)

ggdat.t <- tibble(t=seq(-5,5,length.out=1000))|>
  mutate(pdf.null = dt(t, df=n-1))
# For plotting the observed point
ggdat.obs <- tibble(t = t.stat.a,
                    y = 0) # to plot on x-axis

# Resampling to approximate the sampling distribution
# on the data
R <- 1000
resamples <- tibble(t=numeric(R))
for(i in 1:R){
  curr.sample <- sample(x=x,
                        size=n,
                        replace=T)
  resamples$t[i] = (mean(curr.sample)-mu0)/(sd(curr.sample)/sqrt(n))
}

t.breaks <- c(-5, 0,
              qt(0.95, df = n-1), 14,  # rejection region (right)
              t.stat.a)                  # t-statistic observed
xbar.breaks <- t.breaks * s/(sqrt(n)) + mu0

# Create Plot
ggplot() +
  # null distribution
  geom_line(data=ggdat.t,
            aes(x=t, y=pdf.null))+
  geom_hline(yintercept=0)+
  # rejection region
  geom_ribbon(data=subset(ggdat.t, t>=qt(0.95, df=n-1)),
              aes(x=t, ymin=0, ymax=pdf.null),
              fill="grey", alpha=0.5)+
  # plot p-value (not visible)
  geom_ribbon(data=subset(ggdat.t, t>=t.stat.a),
              aes(x=t, ymin=0, ymax=pdf.null),
              fill="white", alpha=0.25)+
  # plot observation point
  geom_point(data=ggdat.obs, aes(x=t, y=y), color="red")+
  # Resampling Distribution
  stat_density(data=resamples,
               aes(x=t),
               geom="line", color="blue")+
  # clean up aesthetics
  theme_bw()+
  scale_x_continuous("t",
                     breaks = round(t.breaks,2),
                     sec.axis = sec_axis(~.,
                                         name = bquote(bar(x)),
                                         breaks = t.breaks,
                                         labels = round(xbar.breaks,2))) +
  ylab("Density")+
  ggtitle("T Test for Closer Values Compared to Null T Distribution")
```



(b) Question 4, part(b).

```
# b) Farther

mu0 <- 0
x <- fig.2g.data$g.Farther_vals
xbar <- mean(x)
s <- sd(x)
n <- length(x)

ggdat.t <- tibble(t=seq(-5,5,length.out=1000))|>
  mutate(pdf.null = dt(t, df=n-1))
# For plotting the observed point
ggdat.obs <- tibble(t = t.stat.b,
                    y = 0) # to plot on x-axis

# Resampling to approximate the sampling distribution
# on the data
R <- 1000
resamples <- tibble(t=numeric(R))
for(i in 1:R){
  curr.sample <- sample(x=x,
                        size=n,
                        replace=T)
  resamples$t[i] = (mean(curr.sample)-mu0)/(sd(curr.sample)/sqrt(n))
}
```

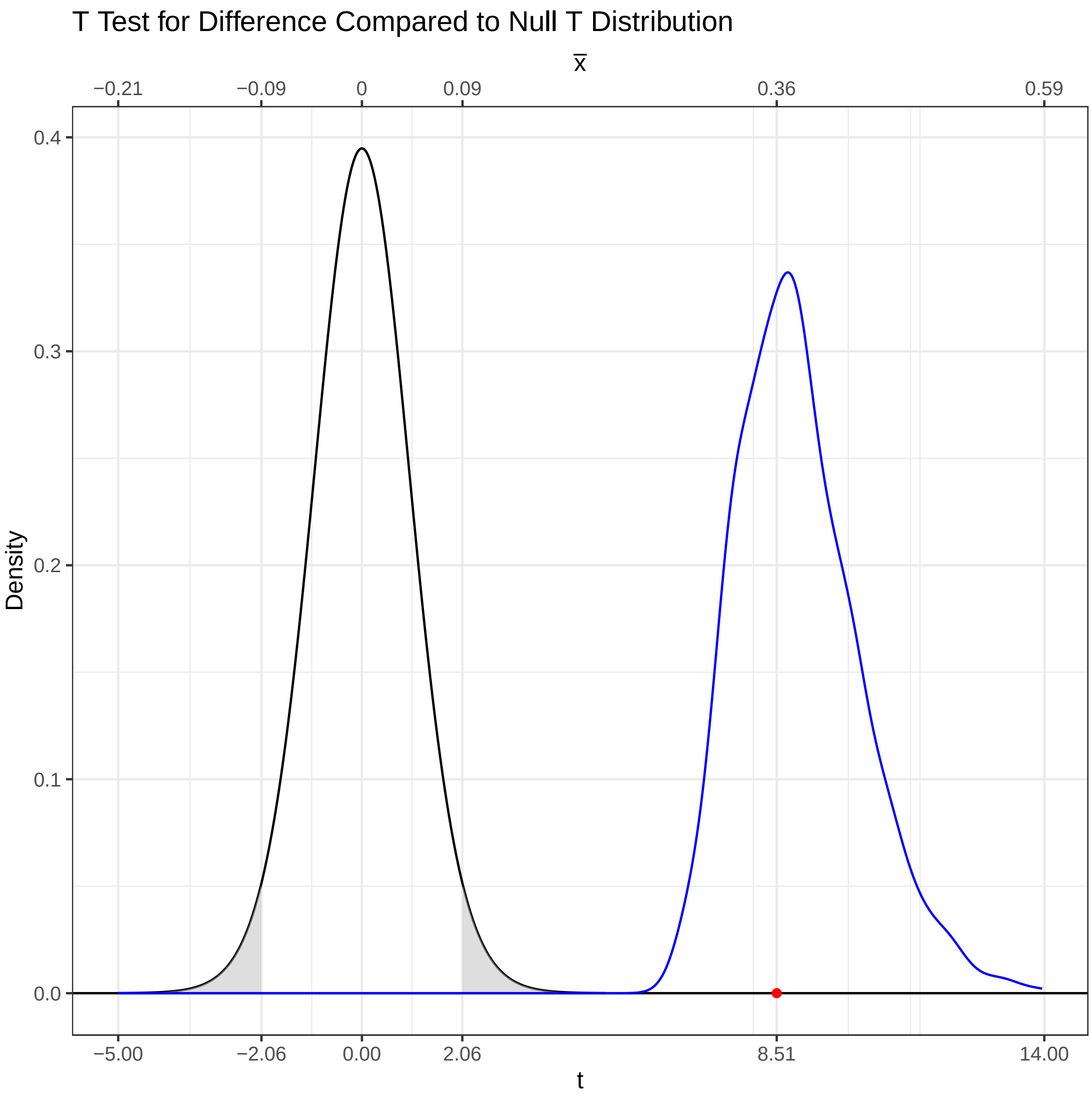
```
t.breaks <- c(-14, qt(0.05, df = n-1), # rejection region (left)
              0, 5,
              t.stat.b)                  # t-statistic observed
xbar.breaks <- t.breaks * s/(sqrt(n)) + mu0

# Create Plot
ggplot() +
  # null distribution
  geom_line(data=ggdat.t,
            aes(x=t, y=pdf.null))+
  geom_hline(yintercept=0)+
  # rejection region
  geom_ribbon(data=subset(ggdat.t, t<=qt(0.05, df=n-1)),
              aes(x=t, ymin=0, ymax=pdf.null),
              fill="grey", alpha=0.5)+
  # plot p-value (not visible)
  geom_ribbon(data=subset(ggdat.t, t>=t.stat.b),
              aes(x=t, ymin=0, ymax=pdf.null),
              fill="white", alpha=0.25)+
  # plot observation point
  geom_point(data=ggdat.obs, aes(x=t, y=y), color="red")+
  # Resampling Distribution
  stat_density(data=resamples,
               aes(x=t),
               geom="line", color="blue")+
  # clean up aesthetics
  theme_bw()+
  scale_x_continuous("t",
                     breaks = round(t.breaks,2),
                     sec.axis = sec_axis(~.,
                                         name = bquote(bar(x)),
                                         breaks = t.breaks,
                                         labels = round(xbar.breaks,2))) +
  ylab("Density")+
  ggtitle("T Test for Farther Values Compared to Null T Distribution")
```



(c) Question 4, part(c).

```
# c) Difference

mu0 <- 0
x <- fig.2g.data$difference
xbar <- mean(x)
s <- sd(x)
n <- length(x)

ggdat.t <- tibble(t=seq(-5,5,length.out=1000))|>
  mutate(pdf.null = dt(t, df=n-1))
# For plotting the observed point
ggdat.obs <- tibble(t = t.stat.c,
                    y = 0) # to plot on x-axis

# Resampling to approximate the sampling distribution
# on the data
R <- 1000
resamples <- tibble(t=numeric(R))
for(i in 1:R){
  curr.sample <- sample(x=x,
                        size=n,
                        replace=T)
  resamples$t[i] = (mean(curr.sample)-mu0)/(sd(curr.sample)/sqrt(n))
}
```

```r
t.breaks <- c(-5, qt(0.025, df = n-1), # rejection region (left)
              0,
              qt(0.975, df = n-1), 14,  # rejection region (right)
              t.stat.c)                  # t-statistic observed
xbar.breaks <- t.breaks * s/(sqrt(n)) + mu0

# Create Plot
ggplot() +
  # null distribution
  geom_line(data=ggdat.t,
            aes(x=t, y=pdf.null))+
  geom_hline(yintercept=0)+
  # rejection region
  geom_ribbon(data=subset(ggdat.t, t<=qt(0.025, df=n-1)),
              aes(x=t, ymin=0, ymax=pdf.null),
              fill="grey", alpha=0.5)+
  geom_ribbon(data=subset(ggdat.t, t>=qt(0.975, df=n-1)),
              aes(x=t, ymin=0, ymax=pdf.null),
              fill="grey", alpha=0.5)+
  # plot p-value (not visible)
  geom_ribbon(data=subset(ggdat.t, t>=t.stat.c),
              aes(x=t, ymin=0, ymax=pdf.null),
              fill="white", alpha=0.25)+
  # plot observation point
  geom_point(data=ggdat.obs, aes(x=t, y=y), color="red")+
  # Resampling Distribution
  stat_density(data=resamples,
               aes(x=t),
               geom="line", color="blue")+
  # clean up aesthetics
  theme_bw()+
  scale_x_continuous("t",
                     breaks = round(t.breaks,2),
                     sec.axis = sec_axis(~.,
                                         name = bquote(bar(x)),
                                         breaks = t.breaks,
                                         labels = round(xbar.breaks,2))) +
  ylab("Density")+
  ggtitle("T Test for Difference Compared to Null T Distribution")
```



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.