1. When conducting the work of Lab 11, we conducted the test that uses the Central Limit Theorem even though the sample size was "small" (i.e., $n < 30$). It turns out, that how "far off" the t-test is can be computed using a first-order Edgeworth approximation for the error. Below, we will do this for the the further observations.

   (a) Boos and Hughes-Oliver (2000) note that

   $$P(T \leq t) \approx F_Z(t) + \underbrace{\frac{\text{skew}}{\sqrt{n}} \frac{(2t^2 + 1)}{6} f_Z(t)}_{\text{error}},$$

   where $f_Z(\cdot)$ and $F_Z(\cdot)$ are the Gaussian PDF and CDF and skew is the skewness of the data. What is the potential error in the computation of the p-value when testing $H_0 : \mu_X = 0; H_a : \mu_X < 0$ using the zebra finch further data?

   ```
   finch.data <- read.csv("zebrafinches.csv")
   n = nrow(finch.data)
   further.sum <- finch.data |>
     summarize(
       mean = mean(further),
       sd = sd(further),
       skew = skewness(further)
     )

   close.t.stat <- t.test(x=finch.data$further, mu = 0, alternative = "less")

   t.stat <- close.t.stat$statistic

   fZ.at.t <- dnorm(t.stat, mean = 0, sd = 1)

   error.val <- ((further.sum$skew)/(sqrt(n)))*(((2*t.stat^2)+1)/6)*fZ.at.t
   error.val
   ```

   ```
   ##            t
   ## -1.226006e-13
   ```

   (b) Compute the error for t statistics from -10 to 10 and plot a line that shows the error across t. Continue to use the skewness and the sample size for the zebra finch further data.

   ```
   t.seq <- seq(-10, 10, length.out=1000)

   fZ <- dnorm(t.seq, mean = 0, sd = 1)

   skew.further <- further.sum$skew

   error.seq <- (skew.further / sqrt(n)) *
     ((2 * t.seq^2 + 1) / 6) *
     fZ

   error.df <- tibble(
     t     = t.seq,
     error = error.seq
   )
   error.df
   ```

   ```
   ## # A tibble: 1,000 x 2
   ##        t      error
   ##    <dbl>      <dbl>
   ##  1 -10    -5.34e-22
   ##  2  -9.98 -6.49e-22
   ##  3  -9.96 -7.90e-22
   ##  4  -9.94 -9.60e-22
   ##  5  -9.92 -1.17e-21
   ##  6  -9.90 -1.42e-21
   ##  7  -9.88 -1.72e-21
   ##  8  -9.86 -2.09e-21
   ##  9  -9.84 -2.53e-21
   ## 10  -9.82 -3.07e-21
   ## # i 990 more rows
   ```

   (c) Suppose we wanted to have a tail probability within 10% of the desired $\alpha = 0.05$. Recall we did a left-tailed test using the further data. How large of a sample size would we need? That is, we

need to solve the error formula equal to 10% of the desired left-tail probability:

$$0.10\alpha \stackrel{set}{=} \underbrace{\frac{\text{skew}}{\sqrt{n}}\frac{(2t^2+1)}{6}f_Z(t)}_{\text{error}},$$

which yields

$$n = \left(\frac{\text{skew}}{6(0.10\alpha)}(2t^2+1)f_Z(t)\right)^2.$$

- **to have a tail probability within 10% of the desired** $\alpha = 0.05$, **we would need a tail-probability of** $3.26e - 07.$

```
alpha = 0.05
assumed.t <- abs(qnorm(alpha))
assumed.fz <- dnorm(assumed.t)
skew.t <-
  tail.prob <- (((skew.further)/6*(0.10*alpha))*((2*assumed.t^2)+1)*assumed.fz)^2
tail.prob

## [1] 3.255548e-07
```

2. Complete the following steps to revisit the analyses from lab 11 using the bootstrap procedure.

   (a) Now, consider the zebra finch data. We do not know the generating distributions for the closer, further, and difference data, so perform resampling to approximate the sampling distribution of the T statistic:
   $$T = \frac{\bar{x}_r - 0}{s/\sqrt{n}},$$
   where $\bar{x}_r$ is the mean computed on the rth resample and s is the sample standard deviation from the original samples. At the end, create an object called `resamples.null.closer`, for example, and store the resamples shifted to ensure they are consistent with the null hypotheses at the average (i.e., here ensure the shifted resamples are 0 on average, corresponding to $t = 0$, for each case).

```
closer <- finch.data$closer

n.closer <- length(closer)
s.closer <- sd(closer)
mu.obs    <- mean(closer)

R = 1000
set.seed(42)

closer.resamples <- tibble(xbar = rep(NA, R), sd = rep(NA, R), t.stat = rep(NA, R))
further <- finch.data$further
difference <- further - closer


further.resamples <- tibble(xbar = rep(NA, R), sd = rep(NA, R), t.stat = rep(NA, R))
diff.resamples <- tibble(xbar = rep(NA, R), sd = rep(NA, R), t.stat = rep(NA, R))
#Generate null bootstrap T-stats
 for (i in 1:R) {
x.star    <- sample(closer, size = n.closer, replace = TRUE)
closer.resamples$xbar[i] <- mean(x.star)
closer.resamples$sd[i] <- sd(x.star)
closer.resamples$t.stat[i] <- (closer.resamples$xbar[i]-mu.obs) / (closer.resamples$sd[i] / sqrt(n.closer))
 }
```

   (b) Compute the bootstrap p-value for each test using the shifted resamples. How do these compare to the t-test p-values? - **The p-values recorded from preforming t-tests for the closer and difference data yielded were very small while the value for the further data was quite high, which makes sense as we are evaluating the left tail of the data. Furthermore, all of the bootstrapped p-values were recorded as** 0, **suggesting that either these values were very small or their measurements were inaccurate.**

```r
# observed T-statistics
t_closer     <- t.test(closer, mu = 0, alternative = "less")$statistic
p_closer_tt <- t.test(closer, mu = 0, alternative = "less")$p.value

t_further    <- t.test(further, mu = 0, alternative = "less")$statistic
p_further_tt <- t.test(further, mu = 0, alternative = "less")$p.value

t_diff       <- t.test(difference, mu = 0, alternative = "less")$statistic
p_diff_tt    <- t.test(difference, mu = 0, alternative = "less")$p.value

#Bootstrap T-stats for further and difference vals
n.further = length(further)
n.diff = length(difference)
mu.further = mean(further)
mu.diff = mean(difference)

for (i in 1:R) {
x.star    <- sample(further, size = n.further, replace = TRUE)
further.resamples$xbar[i] <- mean(x.star)
further.resamples$sd[i] <- sd(x.star)
further.resamples$t.stat[i] <- (further.resamples$xbar[i]-mu.further) / (further.resamples$sd[i] / sqrt(n.further))
}

for (i in 1:R) {
x.star    <- sample(difference, size = n.diff, replace = TRUE)
diff.resamples$xbar[i] <- mean(x.star)
diff.resamples$sd[i] <- sd(x.star)
diff.resamples$t.stat[i] <- (diff.resamples$xbar[i]-mu.diff) / (diff.resamples$sd[i] / sqrt(n.diff))

p_closer_boot  <- mean(abs(closer.resamples$t.stat)  >= abs(t_closer))
p_further_boot <- mean(abs(further.resamples$t.stat) >= abs(t_further))
p_diff_boot    <- mean(abs(diff.resamples$t.stat)    >= abs(t_diff))

p_vals <- tibble(bootstrap_p_vals=(c(p_closer_boot, p_further_boot, p_diff_boot)),
    t_test_p_vals=c(p_closer_tt, p_further_tt, p_diff_tt))
p_vals
}
```

(c) What is the 5th percentile of the shifted resamples under the null hypothesis? - textbfThe 5th percentile of the shifted resamples under the null hypothesis were approximately -1.89 for the closer data, -1.46 for the further data, and -1.64 for the difference data.

```r
 t05_closer_boot   <- quantile(closer.resamples$t.stat,  probs = 0.05)
t05_further_boot <- quantile(further.resamples$t.stat, probs = 0.05)
t05_diff_boot    <- quantile(diff.resamples$t.stat,    probs = 0.05)

# Compute theoretical t critical values at alpha = 0.05 (left tail)
t05_closer_theo   <- qt(0.05, df = n.closer - 1)
t05_further_theo <- qt(0.05, df = n.further - 1)
t05_diff_theo    <- qt(0.05, df = n.diff    - 1)

t05_closer_boot

##      5%
## -1.845572

t05_further_boot

##      5%
## -1.501745

t05_diff_boot

##      5%
## -1.466198
```

Note this value approximates $t_{0.05,n-1}$. Compare these values in each case.

(d) Compute the bootstrap confidence intervals using the resamples. How do these compare to the t-test confidence intervals? - **As displayed within the tibble below, all of the bootstrapped confidence intervals and the t-text intervals were within two decimal places of one another.**

```r
sd.further = sd(further)
sd.diff = sd(difference)
probs <- c(0.025, 0.975)
q_closer       <- quantile(closer.resamples$t.stat,  probs = probs)
ci_closer_boot <- c(
```

3

```
    mean(closer) - q_closer[2] * (s.closer / sqrt(n.closer)),
    mean(closer) - q_closer[1] * (s.closer / sqrt(n.closer))
)
tci_closer      <- t.test(closer,  conf.level = 0.95)$conf.int

# 2. Further data
q_further         <- quantile(further.resamples$t.stat, probs = probs)
ci_further_boot   <- c(
  mean(further) - q_further[2] * (sd.further / sqrt(n.further)),
  mean(further) - q_further[1] * (sd.further / sqrt(n.further))
)
tci_further      <- t.test(further, conf.level = 0.95)$conf.int

# 3. Difference data
q_diff            <- quantile(diff.resamples$t.stat,    probs = probs)
ci_diff_boot      <- c(
  mean(difference) - q_diff[2] * (sd.diff / sqrt(n.diff)),
  mean(difference) - q_diff[1] * (sd.diff / sqrt(n.diff))
)
tci_diff         <- t.test(difference, conf.level = 0.95)$conf.int

ci.tib <- tibble(boot_cis = c(ci_closer_boot, ci_further_boot, ci_diff_boot),
                 t_cis = c(tci_closer, tci_further, tci_diff))
ci.tib

## # A tibble: 6 x 2
##   boot_cis  t_cis
##      <dbl>  <dbl>
## 1    0.120  0.117
## 2    0.198  0.195
## 3   -0.269 -0.257
## 4   -0.156 -0.149
## 5   -0.461 -0.446
## 6   -0.283 -0.272
```

3. Complete the following steps to revisit the analyses from lab 11 using the randomization procedure.

   (a) Now, consider the zebra finch data. We do not know the generating distributions for the closer, further, and difference data, so perform the randomization procedure

```
closer.null  <- closer    - mean(closer)
further.null <- further   - mean(further)
diff.null    <- difference - mean(difference)
rand.t.closer <- replicate(R, {
  signs <- sample(c(-1,1), n.closer, replace = TRUE)
  mean(signs * closer.null)  / (s.closer   / sqrt(n.closer))
})

rand.t.further <- replicate(R, {
  signs <- sample(c(-1,1), n.further, replace = TRUE)
  mean(signs * further.null) / (sd.further / sqrt(n.further))
})

rand.t.diff <- replicate(R, {
  signs <- sample(c(-1,1), n.diff, replace = TRUE)
  mean(signs * diff.null)    / (sd.diff    / sqrt(n.diff))
})
```

   (b) Compute the randomization test p-value for each test.

```
p_closer_rand  <- mean(abs(rand.t.closer)  >= abs(t_closer))
p_further_rand <- mean(abs(rand.t.further) >= abs(t_further))
p_diff_rand    <- mean(abs(rand.t.diff)    >= abs(t_diff))
p_closer_rand

## [1] 0

p_further_rand

## [1] 0

p_diff_rand

## [1] 0
```

   (c) Compute the randomization confidence interval by iterating over values of μ_0.
       Hint: You can "search" for the lower bound from Q_1 and subtracting by 0.0001, and the upper

bound using Q_3 and increasing by 0.0001. You will continue until you find the first value for which the two-sided p-value is greater than or equal to 0.05.

```r
rand_p_val <- function(x, mu0) {
  n     <- length(x)
  s     <- sd(x)
  x0    <- x - mu0
  t_obs <- abs(mean(x) - mu0) / (s / sqrt(n))
  rand_t <- replicate(R, {
    signs <- sample(c(-1,1), n, replace = TRUE)
    mean(signs * x0) / (s / sqrt(n))
  })
  mean(abs(rand_t) >= t_obs)
}
#Loops until it finds the first p value that is less than .05 for both upper and lower bounds
rand_ci <- function(x) {
  m     <- mean(x)
  lower <- m
  while(rand_p_val(x, lower) < 0.05) {
    lower <- lower - step
  }
  upper <- m
  while(rand_p_val(x, upper) < 0.05) {
    upper <- upper + step
  }
  c(lower = lower, upper = upper)
}

ci_closer_rand  <- rand_ci(closer)
ci_further_rand <- rand_ci(further)
ci_diff_rand    <- rand_ci(difference)
ci_closer_rand

##     lower     upper
## 0.1562231 0.1562231

ci_further_rand

##      lower      upper
## -0.2027244 -0.2027244

ci_diff_rand

##      lower      upper
## -0.3589475 -0.3589475
```

4. **Optional Challenge:** In this lab, you performed resampling to approximate the sampling distribution of the T statistic using

$$T = \frac{\bar{x}_r - 0}{s/\sqrt{n}}.$$

I'm curious whether it is better/worse/similar if we computed the statistics using the sample standard deviation of the resamples (s_r), instead of the original sample (s)

$$T = \frac{\bar{x}_r - 0}{s_r/\sqrt{n}}.$$

(a) Perform a simulation study to evaluate the Type I error for conducting this hypothesis test both ways.

(b) Using the same test case(s) as part (a), compute bootstrap confidence intervals and assess their coverage – how often do we 'capture' the parameter of interest?

References

Boos, D. D. and Hughes-Oliver, J. M. (2000). How large does n have to be for z and t intervals? *The American Statistician*, 54(2):121–128.