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?

```
data <- read_csv("zebrafinches.csv")

mu0 <- 0
t.stat <- t.test(data$further,
                    mu = mu0,
                    alternative = "less")
t.further <- t.stat$statistic

n <- length(data$further)

skewness <- skewness(data$further)
fz <- dnorm(t.further)
Fz <- pnorm(t.further)

(edgeworth.error <- (skewness/sqrt(n)) * (((2*(t.further)^2 + 1)/6)*(fz)))

##              t
## -1.226006e-13

(probability <- Fz + edgeworth.error)

##              t
## -1.189164e-13
```

   The probability is $-1.1891637 \times 10^{-13}$ and the Edgeworth approximation is $-1.2260063 \times 10^{-13}$. As the value of the Edgeworth approximation is small, the t-test can be said to be **NOT** "far off".

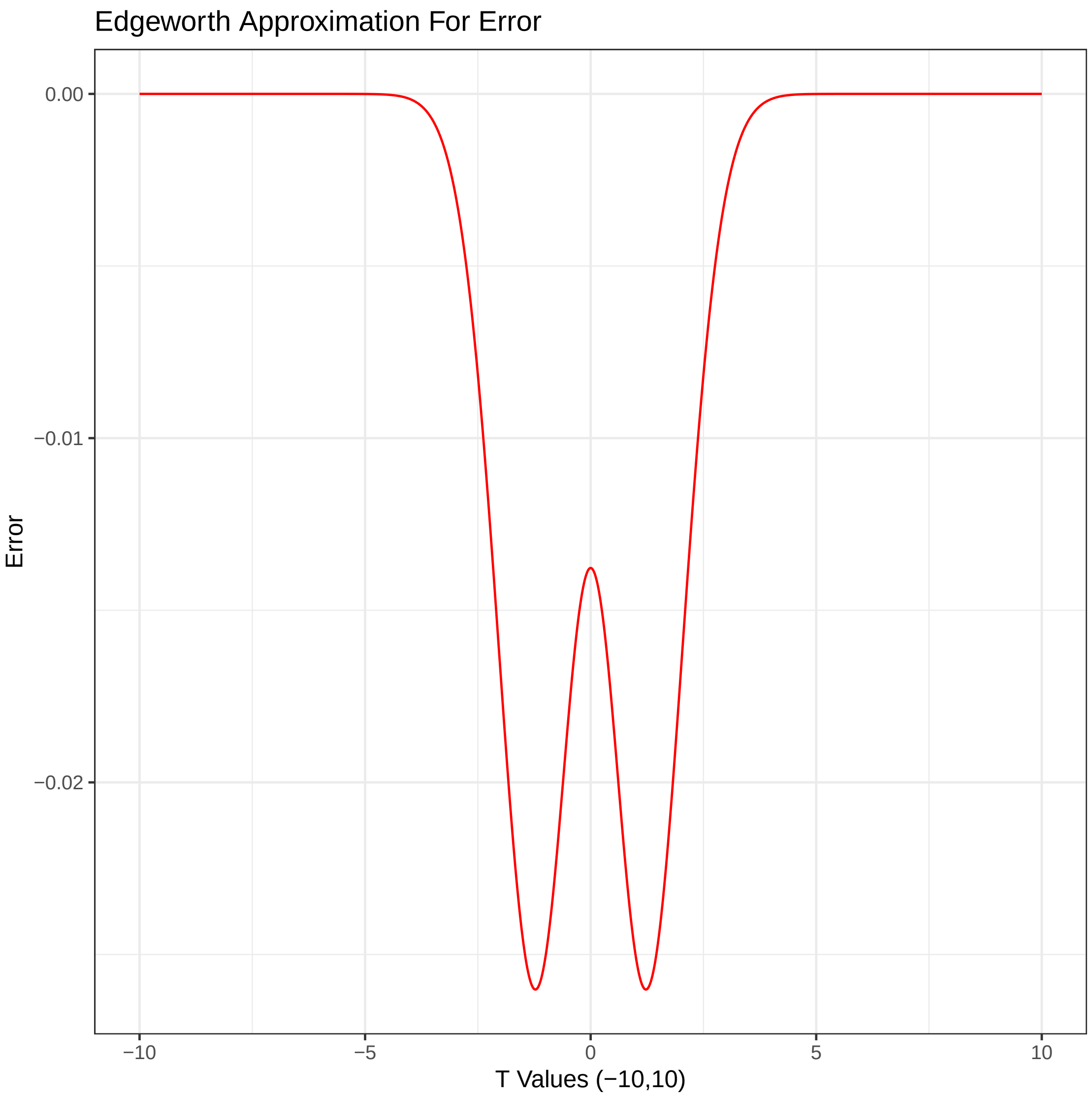
   (b) Compute the error for t statistics from -10 to 10 and plot a line that shows the error across t. Continue to use the skewness and the sample size for the zebra finch further data.

```
t.vals <- seq(-10,10, length = 1000)
fzb <- dnorm(t.vals)

error.vals <- (skewness/sqrt(n)) * (((2*(t.vals)^2 + 1)/6)*(fzb))

tvals.error <- tibble(
  t = t.vals,
  error = error.vals
)

ggplot()+
  geom_line(data = tvals.error,
            aes(x = t, y = error),
            color = "red")+
  theme_bw()+
  labs(title = "Edgeworth Approximation For Error",
       x = "T Values (-10,10)",
       y = "Error")
```



The plot looks like a **W**. The plot shows that t values < 5 and > 5 are zero. The most error

(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:

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

which yields

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

```
val.t <- qnorm(0.05)
fzc <- dnorm(val.t)

(n <- ((skewness/(6*(0.10*0.05))) * (2*val.t^2 + 1) *fzc)^2)

## [1] 520.8876
```

The sample size needed to reach $\alpha = 0.05$ is 520.8876044. To reach 10% of $\alpha = 0.05$, we will need a much larger sample than the 25 taken in the study.

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).

```
R <- 10000
sd.closer <- sd(data$closer)
num.closer <- length(data$closer)
resamples.closer <-  tibble(tstat=rep(NA, R),
                            xbar=rep(NA, R))
for(i in 1:R){
  curr.resample <- sample(x = data$closer,
                          size = num.closer,
                          replace = T)
  resamples.closer$tstat[i] <- (mean(curr.resample)-0)/(sd.closer/sqrt(num.closer))
  resamples.closer$xbar[i] <- mean(curr.resample)
}


closer.delta <- mean(resamples.closer$tstat) - 0

resamples.null.closer <- resamples.closer |>
  mutate(shifted = tstat - closer.delta)

R <- 10000
sd.further <- sd(data$further)
num.further <- length(data$further)
resamples.further <- tibble(tstat=rep(NA, R),
                            xbar=rep(NA, R))
for(i in 1:R){
  curr.resample <- sample(x = data$further,
                          size = num.further,
                          replace = T)
  resamples.further$tstat[i] <- (mean(curr.resample)-0)/(sd.further/sqrt(num.further))
  resamples.further$xbar[i] <- mean(curr.resample)
}


further.delta <- mean(resamples.further$tstat) - 0

resamples.null.further <- resamples.further |>
  mutate(shifted = tstat - further.delta)

R <- 10000
sd.diff <- sd(data$diff)
num.diff <- length(data$diff)
resamples.diff <-  tibble(tstat=rep(NA, R),
                          xbar=rep(NA, R))
for(i in 1:R){
  curr.resample <- sample(x = data$diff,
                          size = num.diff,
                          replace = T)
  resamples.diff$tstat[i] <- (mean(curr.resample)-0)/(sd.diff/sqrt(num.diff))
  resamples.diff$xbar[i] <- mean(curr.resample)
}

diff.delta <- mean(resamples.diff$tstat) - 0
resamples.null.diff <- resamples.diff |>
  mutate(shifted = tstat - diff.delta)

(mean(resamples.null.closer$shifted))

## [1] 1.514344e-16

(mean(resamples.null.further$shifted))

## [1] 1.460165e-16

(mean(resamples.null.diff$shifted))

## [1] 8.837708e-16
```

We tested to see whether the unknown distributions of closer, further, and difference are consistent with the null hypothesis at the average. We used resampling to approximate the distribution and then stored the shifted resamples. The closer shifted mean is $1.5143442 \times 10^{-16}$, further shifted mean is $1.4601653 \times 10^{-16}$, and the difference shifted mean is $8.8377083 \times 10^{-16}$. All of these values are near zero which indicates that the resamples are consistent with the null.

(b) Compute the bootstrap p-value for each test using the shifted resamples. How do these compare to the t-test p-values?

```
# T-Test P-Value
t.p.close <- t.test(data$closer,
                     mu = 0,
                     alternative = "greater")$p.value

t.p.further <- t.test(data$further,
                      mu = 0,
                      alternative = "less")$p.value
```

```
t.p.diff <- t.test(data$diff,
                   mu = 0,
                   alternative = "two.sided")$p.value

# Bootstrap P-Value
boot.p.closer <- mean(resamples.null.closer$shifted >= closer.delta)

boot.p.further <- mean(resamples.null.further$shifted <= further.delta)

low <- 0 - diff.delta
high <- 0 + diff.delta
p.low <- mean(resamples.null.diff$shifted <= low)
p.high <- mean(resamples.null.diff$shifted >= high)
boot.p.diff <- p.low + p.high

(round(t.p.close, 4))

## [1] 0

(boot.p.closer)

## [1] 0

(round(t.p.further, 4))

## [1] 0

(boot.p.further)

## [1] 0

(round(t.p.diff, 4))

## [1] 0

(boot.p.diff)

## [1] 0
```

As can be seen, both the T-Test P-Values and Bootstrap P-Values are all near zero and are represented as zero when rounded to four decimal places.

(c) What is the 5^{th} percentile of the shifted resamples under the null hypothesis? Note this value approximates $t_{0.05,n-1}$. Compare these values in each case.

```
boot.closer.5th <- quantile(resamples.null.closer$shifted, 0.05)
t.closer.5th <- qt(0.05, df = num.closer -1)

boot.further.5th <- quantile(resamples.null.further$shifted, 0.05)
t.further.5th <- qt(0.05, df = num.further -1)

boot.diff.5th <- quantile(resamples.null.diff$shifted, 0.05)
t.diff.5th <- qt(0.05, df = num.diff -1)

(boot.closer.5th)

##      5%
## -1.6016

(t.closer.5th)
```

```
## [1] -1.710882

(boot.further.5th)

##        5%
## -1.689314

(t.further.5th)

## [1] -1.710882

(boot.diff.5th)

##        5%
## -1.551229

(t.diff.5th)

## [1] -1.710882
```

The bootstrapping 5^{th} percentile for the various cases are -1.6015998 for the close, -1.6893145 for the further, and -1.6893145 for the difference. The t-test 5^{th} percentile for the various cases are -1.7108821 for the close, -1.7108821 for the further, and -1.7108821 for the difference. As we can see, the bootstrapping and t-test 5^{th} percentiles are fairly similar to each other.

(d) Compute the bootstrap confidence intervals using the resamples. How do these compare to the *t*-test confidence intervals?

```
boot.CI.closer <- quantile(resamples.null.closer$xbar, c(0.025, 0.975))
t.CI.closer <- t.test(x=data$closer,
                      mu = 0,
                      conf.level = 0.95,
                      alternative = "two.sided")$conf.int

boot.CI.further <- quantile(resamples.null.further$xbar, c(0.025, 0.975))
t.CI.further <- t.test(x=data$further,
                       mu = 0,
                       conf.level = 0.95,
                       alternative = "two.sided")$conf.int

boot.CI.diff <- quantile(resamples.null.diff$xbar, c(0.025, 0.975))
t.CI.diff <- t.test(x=data$diff,
                    mu = 0,
                    conf.level = 0.95,
                    alternative = "two.sided")$conf.int

(boot.CI.closer)

##      2.5%     97.5%
## 0.1201887 0.1926274

(t.CI.closer)

## [1] 0.1173875 0.1950586
## attr(,"conf.level")
## [1] 0.95

(boot.CI.further)
```

```
##       2.5%      97.5% 
## -0.2551750 -0.1557434 
```

```
(t.CI.further)
```

```
## [1] -0.2565176 -0.1489313
## attr(,"conf.level")
## [1] 0.95
```

```
(boot.CI.diff)
```

```
##      2.5%     97.5% 
## 0.2810191 0.4414708 
```

```
(t.CI.diff)
```

```
## [1] 0.2719028 0.4459921
## attr(,"conf.level")
## [1] 0.95
```

As can be seen, the confidence intervals for each case are fairly similar between the bootstrap and t-test and do not differ by a large value.

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

```
randomization.dist <- function(data, mu0, R = 10000) {
  random <- tibble(means = rep(NA, R))

  shifted.data <- data - mu0

  for(i in 1:R) {
    curr.random <- shifted.data * sample(x = c(-1, 1),
                                         size = length(shifted.data),
                                         replace = TRUE)
    random$means[i] <- mean(curr.random)
  }

  random <- random |>
    mutate(means = means + mu0)

  return(random)
}

closer.rand <- randomization.dist(data$closer, 0)
farther.rand <- randomization.dist(data$further, 0)
diff.rand <- randomization.dist(data$diff, 0)

(mean(closer.rand$means))
```

```
## [1] 0.0003950941
```

```
(mean(farther.rand$means))
```

```
## [1] 0.0003820186
```

```
(mean(diff.rand$means))

## [1] -0.0001880823
```

In this step, we found representative distribution for each case. As can be seen, each of the randomized distributions has a mean of about zero.

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

```
closer.mean <- mean(data$closer)
closer.p.random <- mean(closer.rand$means >= closer.mean)

further.mean <- mean(data$further)
further.p.random <- mean(farther.rand$means <= further.mean)

diff.delta <- abs(mean(data$diff) - mu0)
mirror <- mu0 - diff.delta
xbar <- mu0 + diff.delta

diff.p.random <- mean(diff.rand$means <= mirror) +
                 mean(diff.rand$means >= xbar)

(closer.p.random)

## [1] 0

(further.p.random)

## [1] 0

(diff.p.random)

## [1] 0
```

All the p-values are zero which means there is statistically discernible support for the alternative hypothesis. The p-value of zero suggests that the observed mean is in the right tail for close, left tail for further, and in both tails for the difference.

(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.

```
CI.finder <- function(data, R = 1000){

  mu0.iter <- 0.01
  mu0.lower <- mean(data)
  mu0.higher <- mean(data)

  repeat{
    random <- tibble(means = rep(NA, R))

    shifted.data <- data - mu0.lower

    for(i in 1:R) {
      curr.random <- shifted.data * sample(x = c(-1, 1),
                                           size = length(shifted.data),
                                           replace = TRUE)
      random$means[i] <- mean(curr.random)
    }

    random <- random |>
      mutate(means = means + mu0.lower)

    delta <- abs(mean(data) - mu0.lower)
    low <- mu0.lower - delta
    high <- mu0.lower + delta
    p.val <- mean(random$means <= low) +
             mean(random$means >= high)

    if(p.val < 0.05){
      break
    }else{
      mu0.lower <- mu0.lower - mu0.iter
    }
  }

  repeat{
    random <- tibble(means = rep(NA, R))

    shifted.data <- data - mu0.higher

    for(i in 1:R) {
      curr.random <- shifted.data * sample(x = c(-1, 1),
                                           size = length(shifted.data),
                                           replace = TRUE)
      random$means[i] <- mean(curr.random)
    }

    random <- random |>
      mutate(means = means + mu0.higher)

    delta <- abs(mean(data) - mu0.higher)
    low <- mu0.higher - delta
    high <- mu0.higher + delta
    p.val <- mean(random$means <= low) +
      mean(random$means >= high)

    if(p.val < 0.05){
      break
    }else{
      mu0.higher <- mu0.higher + mu0.iter
    }
  }

  return(c(mu0.lower, mu0.higher))
}

(close.CI.randomization <- CI.finder(data$closer))
```

```
## [1] 0.1062231 0.1962231
```

```
(further.CI.randomization <- CI.finder(data$further))

## [1] -0.2627244 -0.1427244

(diff.CI.randomization <- CI.finder(data$diff))

## [1] 0.2589475 0.4489475
```

The confidence intervals are (0.1062231, 0.1962231) for close, (-0.2627244, -0.1427244) for further, and (0.2589475, 0.4489475) for difference. These values are similar to those found earlier in the bootstrap and t-test methods.

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.