Problem Section 3

Two sample inference for means

Jaiden Atterbury

Exercises

1. A group of college students interested in the effect of stepping exercises on the heart conducted an experiment in which subjects were randomly assigned to a stepping exercise on either a low step (coded "low") or a high step (coded "high"). Each subject started with a resting heart rate and performed the exercise for 3 minutes, at which time his or her exercise heart rate was measured. The data are in the file `exercise.csv`.

a. The students first want to see if the random assignment of subjects to the groups did a satisfactory job of equalizing the mean resting heart rates between the groups. Explain how to use a confidence interval to look for evidence of a problem with the random assignment. Then go ahead and implement it.

Before we can go over how we will create a confidence interval to analyze the resting heart rates between the groups let's look at the sample sizes of each group.

Find the sample sizes:

```
exercise %>%
  group_by(Step) %>%
  summarise(sample_size = n())
```

```
## # A tibble: 2 x 2
##   Step  sample_size
##   <chr>       <int>
## 1 high           15
## 2 low            15
```

Let L_i denote the resting heart rate of the ith subject in the low intensity group. Similarly, let H_i denote the resting heart rate of the ith subject in the high intensity group. Since we can think of each group as an independent sample of size 15, we can see that the Central Limit Theorem definitely doesn't apply for these groups. However if we assume that the underlying distribution of each group is normal with differing means but the same variance, namely $L_1, L_2, \ldots, L_{15} \sim Norm(\mu_1, \sigma_0)$ and $H_1, H_2, \ldots, H_{15} \sim Norm(\mu_2, \sigma_0)$, then we can use the t distribution to create a confidence interval to assess the hypotheses $H_0 : \mu_1 = \mu_2$ versus $H_0 : \mu_1 \neq \mu_2$. Hence if we see that 0 is not in the interval we have evidence that the random assignment did not do a satisfactory job of equalizing the mean resting heart rates between the groups.

This t confidence interval will take the form of $\bar{H} - \bar{L} \ \pm \ t_{\alpha/2,n-1} \cdot s_p\sqrt{\frac{1}{n} + \frac{1}{m}}$.

Calculate the interval at the 0.05% level:

```
stats <- exercise %>%
  group_by(Step) %>%
  summarise(sample_size = n(),
            var = var(Resting.HR),
            mean = mean(Resting.HR))
```

```
t_crit <- qt(p = 0.975, df = sum(stats$sample_size) - 2)

s_p <- sqrt((14*stats$var[1] + 14*stats$var[2]) / (sum(stats$sample_size) - 2))

sqt <- sqrt(1/15 + 1/15)

tibble(lower = stats$mean[1] - stats$mean[2] - t_crit * s_p * sqt,
       upper = stats$mean[1] - stats$mean[2] + t_crit * s_p * sqt)
```

```
## # A tibble: 1 x 2
##   lower upper
##   <dbl> <dbl>
## 1 -1.06  12.3
```

As we can see, we fail to reject the null that $\mu_1 = \mu_2$ at the 5% level. Thus we don't have significant evidence that the average resting heart rate between the two groups isn't zero. Thus we can assume that randomization was done properly.

b. Next, you want to test whether there is a difference in heart rates between the two groups. Explain how to use a confidence interval to examine this question. Then go ahead and implement it.

To solve this problem we will add a new column to the dataset that calculates the difference between the heart rate after exercise with the heart rate before exercise. Then we will run a two sample t test to assess the claims.

```
diff_data <- exercise %>%
  mutate(diff = Exercise.HR - Resting.HR)

stats::t.test(diff ~ Step, data = diff_data, var.equal = T)
```

```
##
##  Two Sample t-test
##
## data:  diff by Step
## t = 2.0394, df = 28, p-value = 0.05095
## alternative hypothesis: true difference in means between group high and group low is not equal to 0
## 95 percent confidence interval:
##  -0.05757702 26.05757702
## sample estimates:
## mean in group high  mean in group low
##               32.4               19.4
```

As we can see since the interval contains zero, thus we do not have evidence to say that there is a difference in the change of heart rates between the two groups.

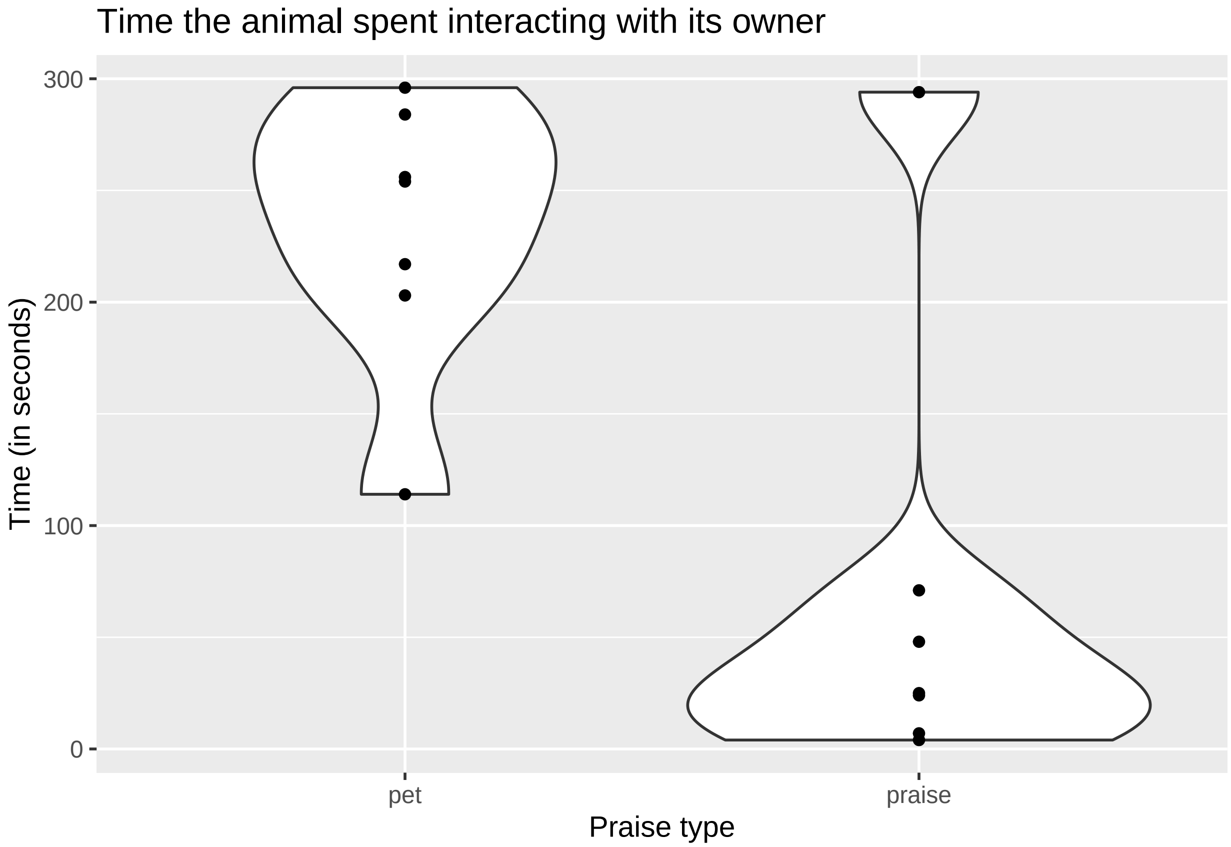
2. For comparing two means using the t-test, we need to assume the data are normally distributed. With small n, it is difficult to judge the shape of a population distribution. Also, if the distribution is skewed the mean is less relevant, but we may still want to test the hypothesis that the two groups have the same distribution. A *permutation test* is a non-parametric method for testing the null hypothesis that the population distributions are identical without specifying their shape.

Consider the following data to determine whether dogs prefer vocal praise or petting. The response variable is the time, in seconds, the animal spent interacting with its owner.

```
dogs <- tibble(
  group = rep(c("pet","praise"), times = c(7,7)),
  times = c(114, 203, 217,254, 256, 284, 296, 4,7, 24, 25, 48, 71, 294)
  )
```

a. Make a violin plot to compare the distributions.

```
ggplot(data = dogs, mapping = aes(x = group, y = times)) +
  geom_violin() +
  geom_point() +
  labs(x = "Praise type",
       y = "Time (in seconds)",
       title = "Time the animal spent interacting with its owner")
```



b. As the true distributions are potentially highly skewed, we focus on the medians. Calculate the observed difference in the medians (pet - praise). This is the observed value of our test statistic. Save it in a variable called `obs_diff_in_median`.

```
med_data <- dogs %>%
  group_by(group) %>%
  summarise(median = median(times))

obs_diff_in_median <- med_data$median[1] - med_data$median[2]
```

c. We test the null hypothesis of identical population distributions against the alternative that the population median is higher for petting using a **permutation test**. The key to permutation testing is this:

if H_0 is true, meaning the population distributions are the same, then if we pool all the data and "reshuffle" the pooled values into two new groups of size $n = 7$ and $m = 7$, we will be sampling from the same (null) distribution.

The following code implements this method once. We set a seed any time random number generation is involved. After dividing the pooled data into two new groups, we re-calculate the difference in the medians.

```
set.seed(1414)


#sample 7 row numbers
i <- sample(1:nrow(dogs), size = 7, replace=FALSE)


new_sample1 <- dogs[i,]$times
new_sample2 <- dogs[-i,]$times

median(new_sample1)-median(new_sample2)
```

```
## [1] -103
```

d. We can generate the sampling distribution of the test statistic by generating *all* possible ways to form the two samples of size n and m from the pooled data, and compute the difference in median for each. Alternately, we could select a subset of the samples randomly and calculate the difference in medians for those. This is the approach we will take here.

The P-value is for the permutation test is the fraction of times the difference between the sample medians is at least `obs_diff_median`.

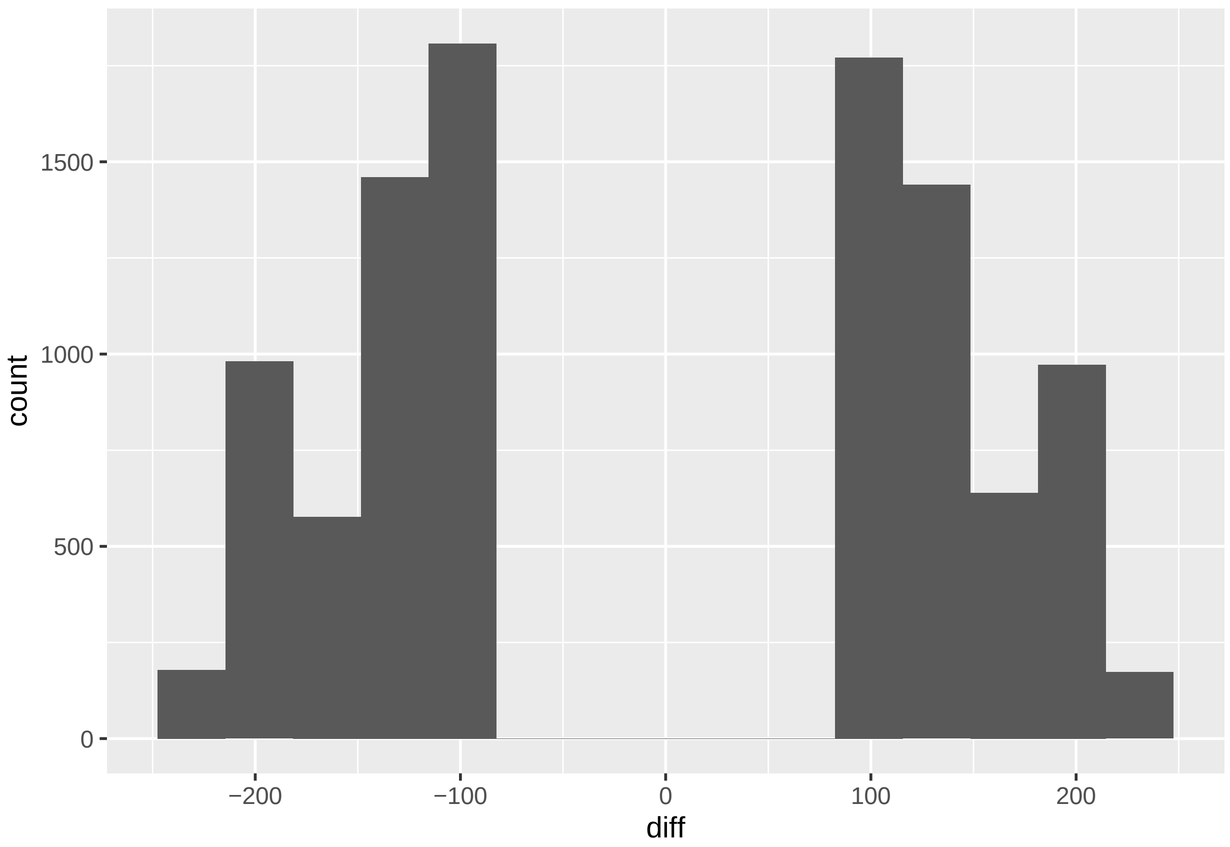
Write code to create B=10,000 new "reshuffles" of the pooled data. For each reshuffle, calculate the difference in sample medians. Then make a plot to visualize the null sampling distribution of this statistic. Finally, calculate the P-value for the permutation test.

```
set.seed(24)
B <- 10000

null_sim_df <- lapply(X = 1:B, FUN = function(X) {
  i = sample(1:nrow(dogs), size = 7, replace=FALSE)
  new_sample1 =  dogs[i,]$times
  new_sample2 = dogs[-i,]$times
  data.frame(diff = median(new_sample1) - median(new_sample2)
)})

med_df <- do.call(rbind, null_sim_df)

ggplot(data = med_df, mapping = aes(x = diff)) +
  geom_histogram(bins = ceiling(log(length(med_df$diff), base=2) + 1))
```



```
2 * (sum(med_df$diff >= obs_diff_in_median) / B)
```

```
## [1] 0.0346
```

The P-values is 0.0346 thus we reject the null that the distributions are the same at the 5% level.