HW 6

SDS348 Spring 2021

Rose Hedderman

This homework is due on Mar 29, 2021 at 8am. Submit a pdf file on Gradescope.

For all questions, include the R commands/functions that you used to find your answer (show R chunk). Answers without supporting code will not receive credit. Write full sentences to describe your findings.

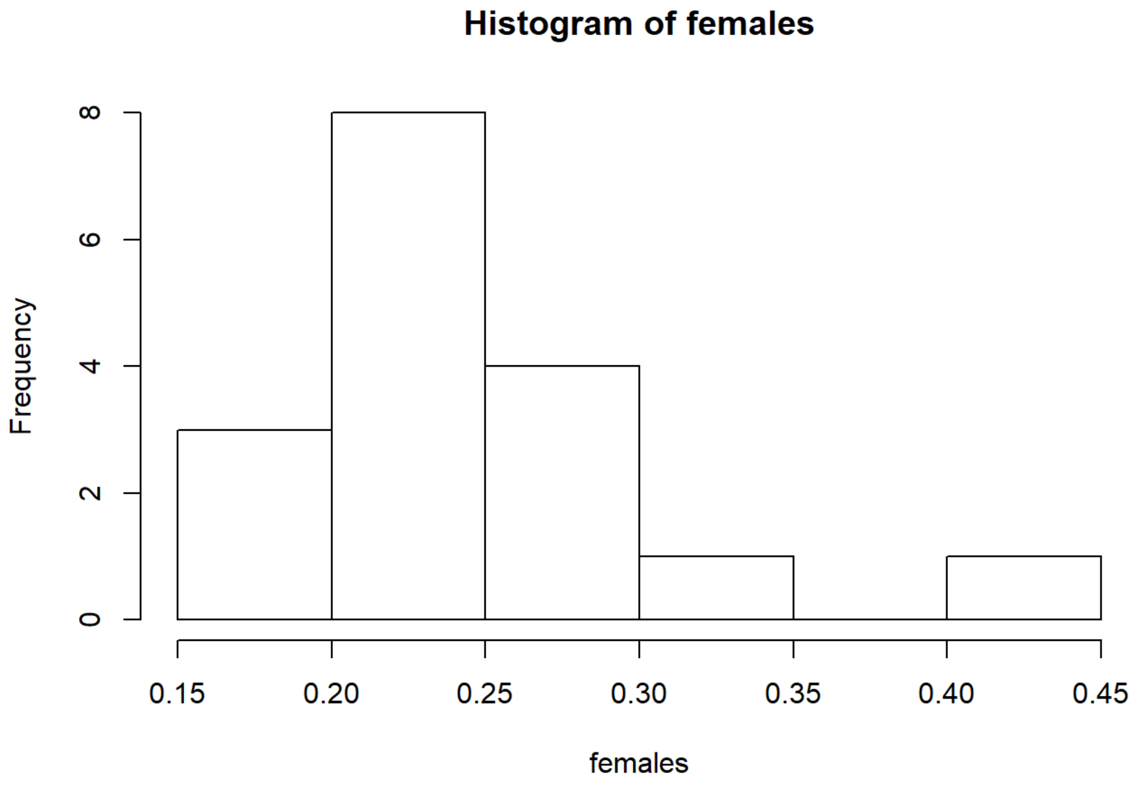
Question 1: (14 pts)

The distribution of mosquito weight for the Aedes aegypti species is known to be log-normal (that is, weight is normally distributed if transformed with the natural log). Untransformed weights of 17 female and 15 male mosquitoes are given below (mg).

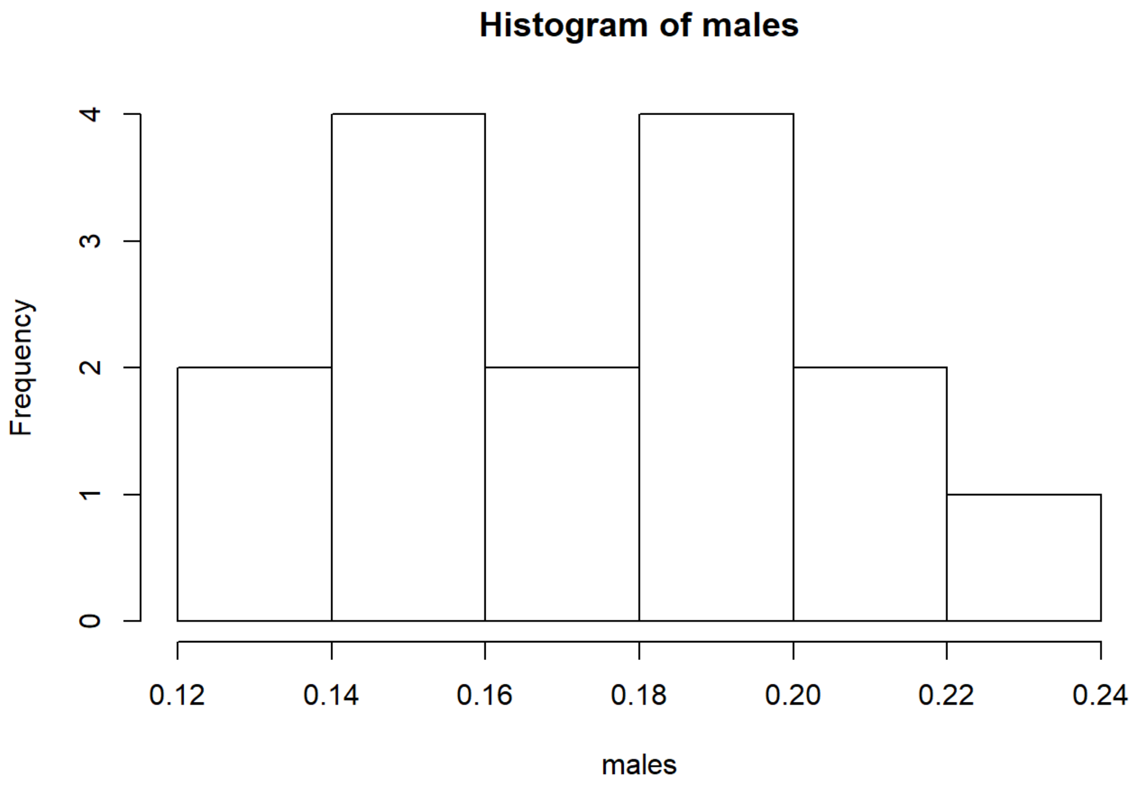
Females: 0.291, 0.208, 0.241, 0.437, 0.228, 0.256, 0.208, 0.234, 0.280, 0.340, 0.150, 0.211, 0.168, 0.221, 0.237, 0.189, 0.261
Males: 0.185, 0.222, 0.149, 0.187, 0.191, 0.219, 0.132, 0.144, 0.140, 0.159, 0.172, 0.198, 0.154, 0.201, 0.167

1.1 (2 pts) Represent the distribution of weights for females and for males in a histogram (you can use the function `hist()` to make simple histograms). Do a log transformation of weights for females and for males. Represent the transformed distributions in simple histograms. Has the log transformation improved the normality assumption?

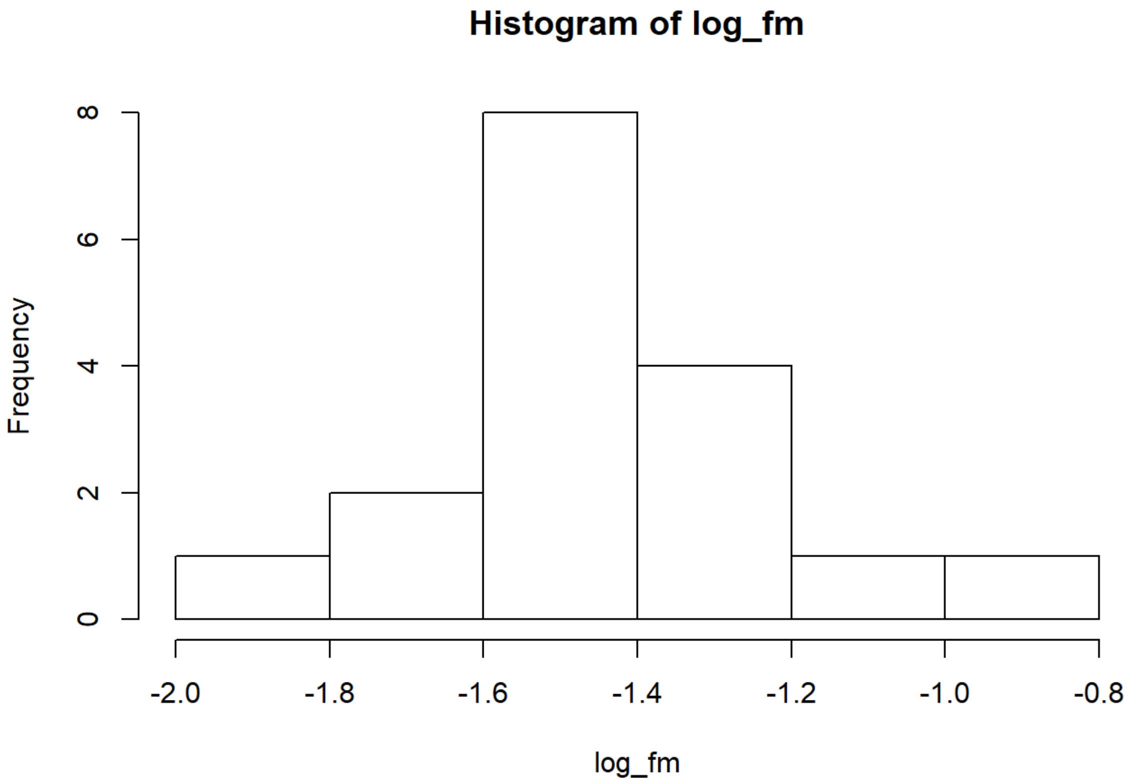
```
females <- c(0.291, 0.208, 0.241, 0.437, 0.228, 0.256, 0.208, 0.234, 0.280, 0.340,
             0.150, 0.211, 0.168, 0.221, 0.237, 0.189, 0.261)
males <- c(0.185, 0.222, 0.149, 0.187, 0.191, 0.219, 0.132, 0.144, 0.140, 0.159,
           0.172, 0.198, 0.154, 0.201, 0.167)

# original female histogram
hist(females)
```



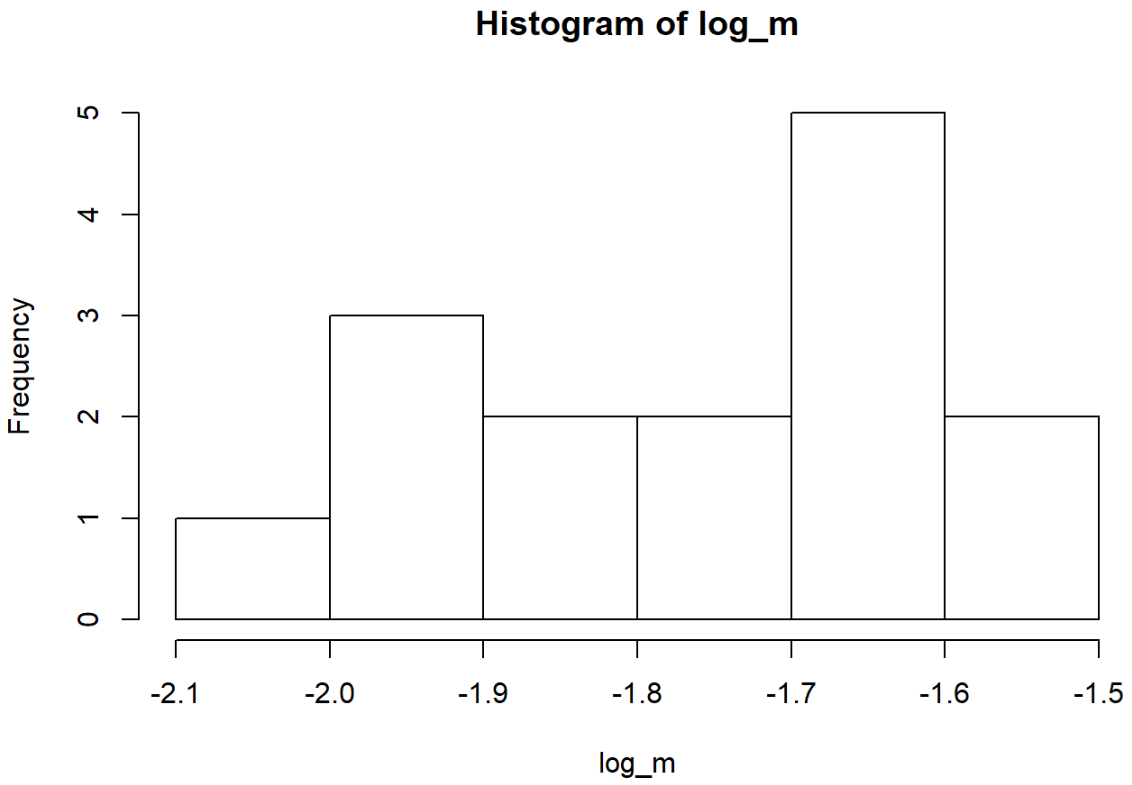
```
# original male histogram
hist(males)
```



```
# transform male and female datasets
log_fm <- log(females)
log_m <- log(males)
# transformed female histogram
hist(log_fm)
```



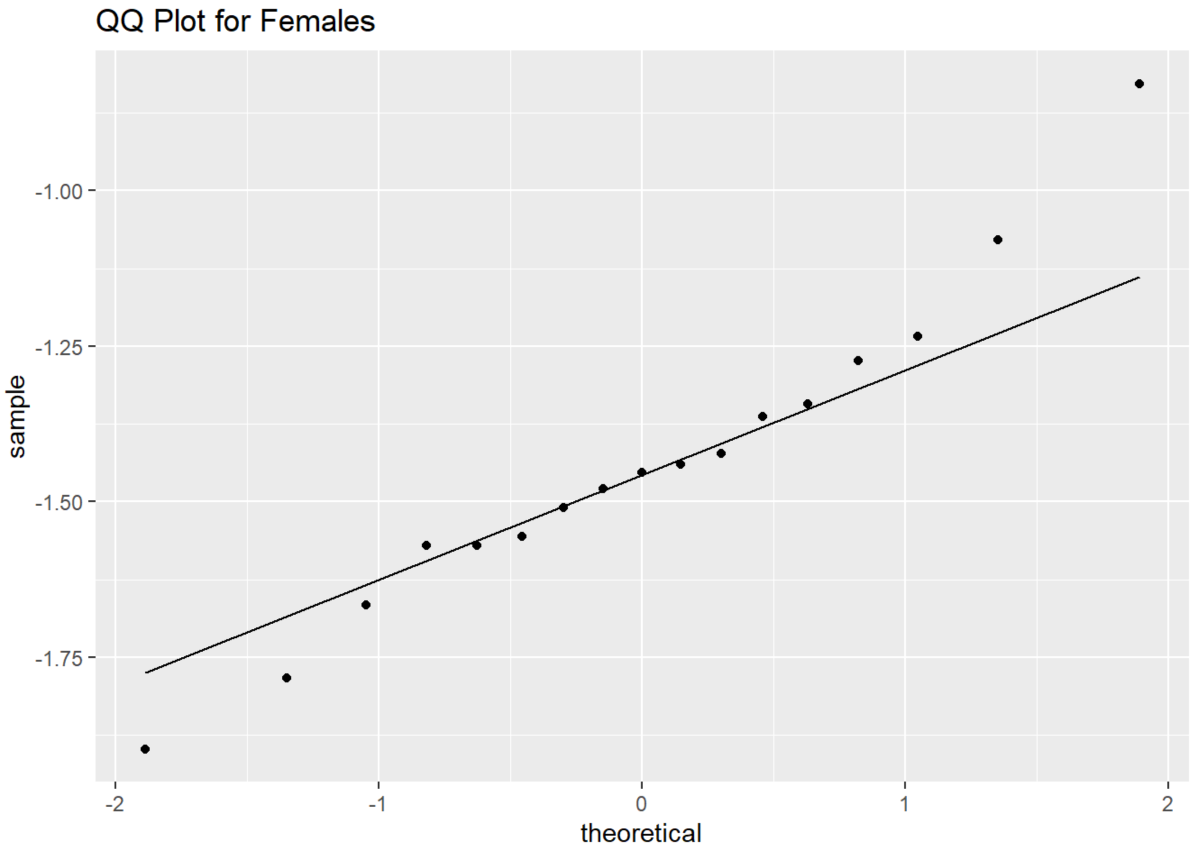
```
# transformed male histogram
hist(log_m)
```



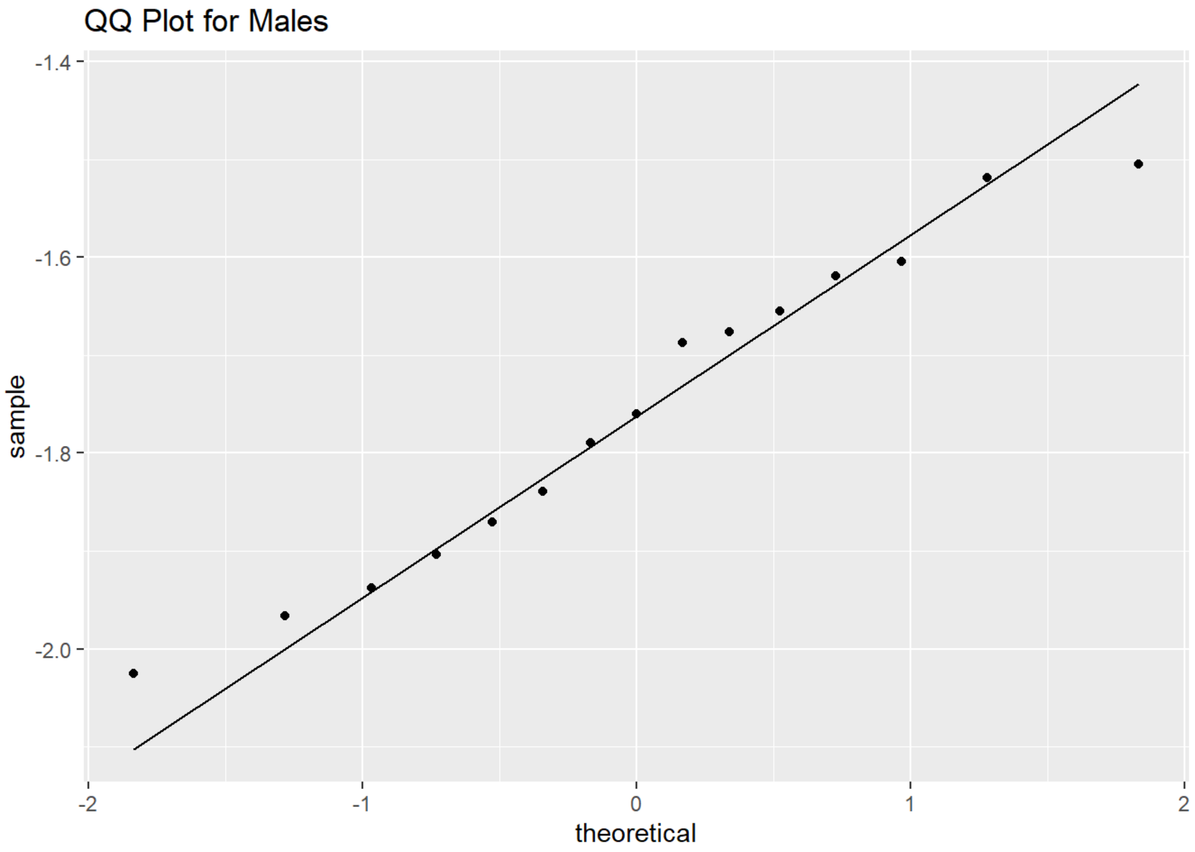
The log transformation has improved the normality assumption of the female and male dataset, but improved the female dataset more.

1.2 (2 pts) Do the two groups weigh the same on average? We would like to perform an independent t-test. Assuming the samples were random and the observations were independent, check the rest of the assumptions (construct QQ-plots, conduct appropriate tests).

```
# construct QQ plot for normality assumption for females
data.frame(log_fm) %>%
  ggplot(aes(sample = log_fm)) +
  # Compare quantiles (by default = standard normal)
  stat_qq() +
  # Reference Q-Q line
  stat_qq_line() +
  labs(title = "QQ Plot for Females")
```


QQ Plot for Females

```
# construct QQ plot for normality assumption for males
data.frame(log_m) %>%
  ggplot(aes(sample = log_m)) +
  # Compare quantiles (by default = standard normal)
  stat_qq() +
  # Reference Q-Q line
  stat_qq_line() +
  labs(title = "QQ Plot for Males")
```



```
# perform shapiro wilk test for p value
# male shapiro test
shapiro.test(log_m)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  log_m
## W = 0.96151, p-value = 0.7186
```

```
# female shapiro test
shapiro.test(log_fm)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  log_fm
## W = 0.96655, p-value = 0.7556
```

The qq-plot shows that the log transformation of the datasets are normal. The p value of the male log transformation is 0.7186 which determines that we fail to reject the null hypothesis. The p value of the female log transformation is 0.7556 which determines that we fail to reject the null hypothesis..

1.3 (4 pts) After verifying the assumptions, perform the appropriate t-test. Write the hypotheses and write a conclusion in context, citing the appropriate statistics.

```
# t test comparing the two
t.test(log_fm, log_m,var.equal = FALSE)
```

```
##
##  Welch Two Sample t-test
##
## data:  log_fm and log_m
## t = 4.2722, df = 27.664, p-value = 0.0002059
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.1655087 0.4707463
## sample estimates:
## mean of x mean of y
## -1.439187 -1.757315
```

The two sample t-test has a p-value of 0.0002 which determines that we reject the null hypothesis.

1.4 (1 pt) Build a dataframe called `mosquitoes` with a column for `weight` , a column for `logweight` , and a column for `sex` . What are the observed difference of the mean weights for untransformed and transformed data? Call them `obs_diff` and `obs_logdiff` respectively.

```
# build dataframe mosquitoes
mosquitoes <- data.frame(
  # with column weight
  weight = c(females,males),
  # with column logweight
  logweight = c(log_fm,log_m),
  # with column sex
  sex = c(rep("females",17), rep("males",15)))
head(mosquitoes)
```

```
##   weight  logweight     sex
## 1  0.291 -1.2344320 females
## 2  0.208 -1.5702172 females
## 3  0.241 -1.4229583 females
## 4  0.437 -0.8278221 females
## 5  0.228 -1.4784097 females
## 6  0.256 -1.3625778 females
```

```
# observed difference of mean weights for untransformed data
obs_diff <- mosquitoes %>%
  group_by(sex) %>%
  summarize(means = mean(weight)) %>%
  summarize(mean_diff = diff(means)) %>%
  pull
obs_diff
```

```
## [1] -0.07003922
```

```
# observed difference of mean weights for transformed data
obs_logdiff <- mosquitoes %>%
  group_by(sex) %>%
  summarize(means = mean(logweight)) %>%
  summarize(mean_diff = diff(means)) %>%
  pull
obs_logdiff
```

```
## [1] -0.3181275
```

The observed difference of the mean weights for the untransformed data was -0.07 and the observed difference of the mean weights for the transformed data was -0.3181.

1.5 (3 pts) After setting the seed as specified below, perform a randomization test on the original weight data *then* on the log weight data. That is, for both, generate a distribution of 5000 mean differences on randomized data (with a `for` loop, *note: it might take some time to run*). Compute and report two-tailed p-values in both cases. Do both randomization tests agree? What does it mean? Are your conclusions the same as they were above for the parametric t-test?

```
set.seed(348)
mean_diff <- vector()
# randomization test on the original weight data
for(i in 1:5000){
  temp <- data.frame(sex = mosquitoes$sex, weight = sample(mosquitoes$weight))

  mean_diff[i] <- temp %>%
    group_by(sex) %>%
    summarize(means = mean(weight)) %>%
    summarize(mean_diff = diff(means)) %>%
    pull
}
mean_diff2 <- vector()
# randomization test on the log weight data
for(i in 1:5000){
  temp <- data.frame(sex = mosquitoes$sex, logweight = sample(mosquitoes$logweight))

  mean_diff2[i] <- temp %>%
    group_by(sex) %>%
    summarize(means = mean(logweight)) %>%
    summarize(mean_diff = diff(means)) %>%
    pull
}
# T test for original weight data
t.test(weight ~ sex, data = mosquitoes, var.equal = T)
```

```
##
##  Two Sample t-test
##
## data:  weight by sex
## t = 3.7368, df = 30, p-value = 0.0007829
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.0317604 0.1083180
## sample estimates:
## mean in group females   mean in group males
##             0.2447059             0.1746667
```

```
# T test for logweight data
t.test(logweight ~ sex, data = mosquitoes, var.equal = T)
```

```
## 
##  Two Sample t-test
## 
## data:  logweight by sex
## t = 4.1606, df = 30, p-value = 0.0002452
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.1619712 0.4742838
## sample estimates:
## mean in group females   mean in group males 
##             -1.439187             -1.757315
```

The p-value for the original weight data is 0.0007 while the p-value for the logweight data is 0.0002. This means that both the original and the logweight data have sufficient evidence to reject the null hypothesis.

1.6 (2 pts) Recall the observed mean difference in mosquito weights between the two groups (calculated for question 4.4). Now you will create a 95% CI for this difference in means using bootstrapping. Resample from the original male mosquito data with replacement using `sample(..., replace=T)` , resample from the original female mosquito data with replacement with `sample(..., replace=T)` , take the mean difference of these samples, save it, and repeat this process 5000 times (with a `for` loop). What is the mean of the resulting distribution? Report the 95% CI of this distribution by reporting the .025 and the 0.975 percentiles of mosquito weight differences. Interpret it in a sentence.

```
set.seed(348)

diffmeans <- vector()

for (i in 1:5000){
  # Resample from the original male mosquito data with replacement
  m_sample <- sample(males,replace = T)
  # resample from the original female mosquito data with replacement
  fm_sample <- sample(females, replace = T)
  # take the mean difference of these samples
  diffmeans[i] <- mean(fm_sample) - mean(m_sample)
}
mean(diffmeans)
```

```
## [1] 0.07021834
```

```
quantile(diffmeans,c(.025, .975))
```

```
##       2.5%      97.5% 
## 0.03879186 0.10634118
```

The mean of diffmeans (the resulting distribution is 0.0702. According to the confidence interval, there is a 95% confidence that the mean difference in mosquito weights is between 0.03879 and 0.10634.

Question 2: (11 pts)

For this question, we will use the pottery data set which contains the chemical composition (the percentage of metal oxide or abundance) of ancient pottery found at four sites in Great Britain.

2.1 (0.5 pt) Import the dataset from an online resource. How many rows and how many columns are in this dataset? What does a row represent? What does a column represent?

```
pottery <- read_csv("https://wilkelab.org/classes/SDS348/data_sets/pottery.csv")
# preview pottery
head(pottery)
```

```
## # A tibble: 6 x 6
##   Site         Al    Fe    Mg    Ca    Na
##   <chr>     <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Llanedyrn  14.4  7     4.3   0.15  0.51
## 2 Llanedyrn  13.8  7.08  3.43  0.12  0.17
## 3 Llanedyrn  14.6  7.09  3.88  0.13  0.2
## 4 Llanedyrn  11.5  6.37  5.64  0.16  0.14
## 5 Llanedyrn  13.8  7.06  5.34  0.2   0.2
## 6 Llanedyrn  10.9  6.26  3.47  0.17  0.22
```

```
# number of rows in pottery
nrow(pottery)
```

```
## [1] 26
```

```
# number of columns in pottery
ncol(pottery)
```

```
## [1] 6
```

In the pottery dataset, there are 26 rows and 6 columns. Each row represents a datapoint and each column represents a type of variable recorded.

2.2 (2.5 pts) Let’s compare the chemical composition of aluminium (Al) across the different sites. Compute the SSB (sum of squares between groups) and SSW (sum of squares within groups) for a one-way ANOVA, manually (use `dplyr` functions to get group means, finding the sum of the differences squared, …). Use the calculated values of SSB and SSW to compute an F statistic. Use `pf(..., df1=, df2=, lower.tail=F)` on the F statistic you calculated to determine the p-value. Compare your results to the output from `summary(aov())`. What is your conclusion about the chemical composition of aluminium across sites?

```
# Compute the SSW (sum of squares within groups)
# for a one-way ANOVA
# manually (use `dplyr` functions to get group means
# finding the sum of the differences squared
SSW <- pottery %>%
    group_by(Site) %>%
    summarize(SSW = sum((Al - mean(Al))^2)) %>%
    summarize(sum(SSW)) %>%
    pull
  # Compute variation between groups
# Compute the SSB (sum of squares between groups)
# for a one-way ANOVA
# manually (use `dplyr` functions to get group means
# finding the sum of the differences squared
SSB <- pottery %>%
    mutate(mean = mean(Al)) %>%
    group_by(Site) %>%
    mutate(groupmean = mean(Al)) %>%
    summarize(SSB = sum((mean - groupmean)^2)) %>%
    summarize(sum(SSB)) %>%
    pull
# Use the calculated values of SSB and SSW to compute an F statistic
  # df for SSB is 4 groups - 1 = 3
  # df for SSW is 26 observations - 4 groups = 22
Fs <- (SSB/3)/(SSW/22)
# determine the p-value
pf(Fs,3,22,lower.tail = F)
```

```
## [1] 1.62687e-07
```

```
# Compare your results to the output
summary(aov(Al ~ Site, data = pottery))
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)    
## Site         3 175.61   58.54   26.67 1.63e-07 ***
## Residuals   22  48.29    2.19                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Your answer goes here. 1-2 sentences.

2.3 (1 pt) Let’s conduct a MANOVA test to investigate whether at least one of these five response variables (chemical compositions) differ by site. Use `manova(cbind(Y1,Y2,Y3...)~X,data=data)` and report the results in writing. *Don’t worry about the assumptions (there are lots).*

```
# conduct a MANOVA test
manova_pot <- manova(cbind(Al, Fe,Mg,Ca,Na)~Site,data=pottery)

# OUtput of MANOVA
summary(manova_pot)
```

```
##           Df Pillai approx F num Df den Df    Pr(>F)    
## Site       3 1.5539   4.2984     15     60 2.413e-05 ***
## Residuals 22                                            
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Your answer goes here. 1-2 sentences.

2.4 (2 pts) Now, let’s investigate which of the elements differ by site. Report full ANOVA results for each metal variable. For the ones that differ, explore which sites are different, that is, perform posthoc t-tests for all significant ANOVAs using `pairwise.t.test(...,p.adj="none")` (*you do not have to write anything up about the post hoc tests for now*).

```
# Report full ANOVA results for each metal variable
summary.aov(manova_pot)
```

```
##  Response Al :
##             Df  Sum Sq Mean Sq F value    Pr(>F)
## Site         3 175.610  58.537  26.669 1.627e-07 ***
## Residuals   22  48.288   2.195
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response Fe :
##             Df  Sum Sq Mean Sq F value    Pr(>F)
## Site         3 134.222  44.741  89.883 1.679e-12 ***
## Residuals   22  10.951   0.498
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response Mg :
##             Df Sum Sq Mean Sq F value    Pr(>F)
## Site         3 103.35  34.450   49.12 6.452e-10 ***
## Residuals   22  15.43   0.701
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response Ca :
##             Df   Sum Sq  Mean Sq F value    Pr(>F)
## Site         3 0.204703 0.068234  29.157 7.546e-08 ***
## Residuals   22 0.051486 0.002340
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response Na :
##             Df  Sum Sq  Mean Sq F value    Pr(>F)
## Site         3 0.25825 0.086082  9.5026 0.0003209 ***
## Residuals   22 0.19929 0.009059
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# perform posthoc t-tests for all significant ANOVAs
# If ANOVA is significant then we can perform post-hoc analysis
    # For Al
pairwise.t.test(pottery$Al,pottery$Site, p.adj="none")
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  pottery$Al and pottery$Site
##
##            AshleyRails Caldicot IsleThorns
## Caldicot   0.00016     -        -
## IsleThorns 0.36866     3.0e-05  -
## Llanedyrn  3.3e-06     0.44848  2.7e-07
##
## P value adjustment method: none
```

```
    # For Fe
pairwise.t.test(pottery$Fe,pottery$Site, p.adj="none")
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  pottery$Fe and pottery$Site
##
##            AshleyRails Caldicot IsleThorns
## Caldicot   1.2e-06     -        -
## IsleThorns 0.658       2.6e-06  -
## Llanedyrn  6.0e-12     0.086    1.4e-11
##
## P value adjustment method: none
```

```
    # For Mg
pairwise.t.test(pottery$Mg,pottery$Site, p.adj="none")
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  pottery$Mg and pottery$Site
##
##            AshleyRails Caldicot IsleThorns
## Caldicot   0.00013     -        -
## IsleThorns 0.89901     0.00016  -
## Llanedyrn  2.2e-09     0.13917  2.9e-09
##
## P value adjustment method: none
```

```
    # For Ca
pairwise.t.test(pottery$Ca,pottery$Site, p.adj="none")
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  pottery$Ca and pottery$Site
##
##            AshleyRails Caldicot IsleThorns
## Caldicot   4.8e-06     -        -
## IsleThorns 0.405       1.1e-06  -
## Llanedyrn  5.4e-06     0.019    5.2e-07
##
## P value adjustment method: none
```

```
    # For Na
pairwise.t.test(pottery$Na,pottery$Site, p.adj="none")
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  pottery$Na and pottery$Site
##
##            AshleyRails Caldicot IsleThorns
## Caldicot   0.98019     -        -
## IsleThorns 0.92150     0.96039  -
## Llanedyrn  0.00049     0.01068  0.00065
##
## P value adjustment method: none
```

According to the full ANOVA results for each variable, each of the 5 variables reported a value lower than 0.05 therefore we can reject the null hypothesis and the results are considered significant. Because of this, 5 post-hoc analyses were run.

2.5 (2 pts) Between 2.3 and 2.4, how many hypothesis tests have you done in total? What is the probability that you have made at least one type I error (i.e., what is the overall type-I error rate)? What (Bonferroni adjusted) significance level should you use if you want to keep the overall type I error rate at .05? Which of your post hoc tests that were significant before the adjustment are no longer significant?

```
# probability that you have made at least one type I error
prob <- 1 - (.95^11)
prob
```

```
## [1] 0.4311999
```

```
# Bonferroni adjusted
bon_adj <- .05/11
bon_adj
```

```
## [1] 0.004545455
```

Between 2.3 and 2.4, 11 hypothesis tests have been done in total. The probability that you have made at least one type I error is 0.4312 and the Bonferroni adjusted value is 0.004. None of the post-hoc analyses are no longer significant.

2.6 (1 pt) Let’s now conduct a PERMANOVA test. Calculate the distances between each metal and each pot in the pottery dataset, using the function `dist` . Use the `adonis()` function from the `vegan` package to conduct PERMANOVA. Is the p-value larger or smaller than in the parametric MANOVA? Why might that be?

```
library(vegan)
# Calculate the distances between each metal and each pot in the pottery dataset
dist_Al <- pottery %>%
  select(Al, Site) %>%
  dist
# conduct a PERMANOVA test
adonis(dist_Al ~ Site, data = pottery)
```

```
##
## Call:
## adonis(formula = dist_Al ~ Site, data = pottery)
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## Site       3    351.22  117.07  26.669 0.78433  0.001 ***
## Residuals 22     96.58    4.39         0.21567
## Total     25    447.80                 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
dist_Fe <- pottery %>%
  select(Fe, Site) %>%
  dist
# conduct a PERMANOVA test
adonis(dist_Fe ~ Site, data = pottery)
```

```
##
## Call:
## adonis(formula = dist_Fe ~ Site, data = pottery)
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## Site       3   268.443  89.481  89.883 0.92457  0.001 ***
## Residuals 22    21.902   0.996         0.07543
## Total     25   290.345                 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
dist_Mg <- pottery %>%
  select(Mg, Site) %>%
  dist
# conduct a PERMANOVA test
adonis(dist_Mg ~ Site, data = pottery)
```

```
##
## Call:
## adonis(formula = dist_Mg ~ Site, data = pottery)
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model     R2 Pr(>F)
## Site       3   206.701  68.900   49.12 0.8701  0.001 ***
## Residuals 22    30.859   1.403         0.1299
## Total     25   237.560                 1.0000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
dist_Ca <- pottery %>%
  select(Ca, Site) %>%
  dist
# conduct a PERMANOVA test
adonis(dist_Ca ~ Site, data = pottery)
```

```
##
## Call:
## adonis(formula = dist_Ca ~ Site, data = pottery)
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs  MeanSqs F.Model      R2 Pr(>F)
## Site       3   0.40941 0.136468  29.157 0.79903  0.001 ***
## Residuals 22   0.10297 0.004681         0.20097
## Total     25   0.51238                  1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
dist_Na <- pottery %>%
  select(Na, Site) %>%
  dist
# conduct a PERMANOVA test
adonis(dist_Na ~ Site, data = pottery)
```

```
##
## Call:
## adonis(formula = dist_Na ~ Site, data = pottery)
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs  MeanSqs F.Model      R2 Pr(>F)
## Site       3   0.51649 0.172164  9.5026 0.56442  0.004 **
## Residuals 22   0.39859 0.018118         0.43558
## Total     25   0.91508                  1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

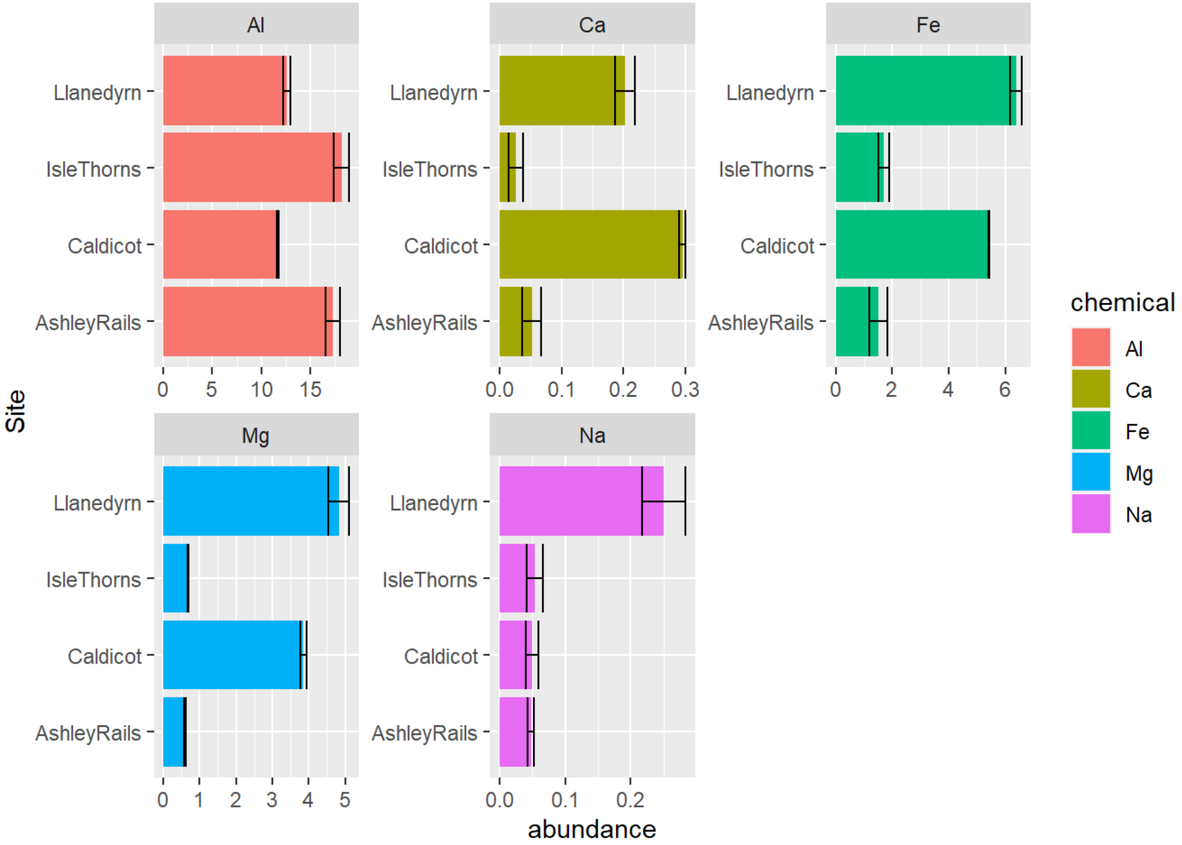
The p-value is larger than the p-value of the MANOVA possibly because the PERMANOVA is less sensitive.

2.7 (2 pts) Make the pottery dataset long by pivoting all of the element names into a column with all of the corresponding values into another column. Use that data to make a plot (mapping abundance to x, site to y), showing the average abundance of each element at each site with `geom_bar()` (using `stat = "summary", fun = "mean"`), adding standard errors (with `geom_errorbar(stat = "summary", fun.data = "mean_se")`) and then faceting by element (set scales='free'). (Add bootstrapped with `geom_errorbar(stat="summary",fun.data=mean_cl_boot)` , or by computing them manually. *Hint: refer to HW2 for similar graphs.* For which element there is the most noticeable difference between one location compared to the others?

```
#library("Hmisc")
new_pottery <- pottery %>%
  pivot_longer(c('Al','Fe','Mg','Ca','Na'), names_to = "chemical", values_to = "abundance")
head(new_pottery)
```

```
## # A tibble: 6 x 3
##   Site      chemical abundance
##   <chr>     <chr>        <dbl>
## 1 Llanedyrn Al           14.4
## 2 Llanedyrn Fe            7
## 3 Llanedyrn Mg            4.3
## 4 Llanedyrn Ca            0.15
## 5 Llanedyrn Na            0.51
## 6 Llanedyrn Al           13.8
```

```
ggplot(new_pottery, aes(abundance,Site, fill = chemical)) +
  geom_bar(stat= "summary", fun = "mean") +
  geom_errorbar(stat="summary", fun.data = "mean_se")+
  facet_wrap(vars(chemical), scales = "free")
```



```
# + geom_errorbar(stat="summary", fun.data = mean_cl_boot())
```

Your answer goes here. 1-2 sentences.

```
##        sysname        release        version       nodename        machine
##      "Windows"       "10 x64"  "build 19042"     "ROSE-XPS"       "x86-64"
##          login           user effective_user
##        "roseh"        "roseh"        "roseh"
```