HW5: Chapter 3, Section 5.5, Section 6.3, Section 6.4

Solutions

The code below just loads some packages and makes it so that enough digits are printed that you won't get confused by rounding errors.

```
library(dplyr) # functions like summarize
library(ggplot2) # for making plots
library(mosaic) # convenient interface to t.test function
library(readr)
library(gmodels)

options("pillar.sigfig" = 10) # print 10 significant digits in summarize output
```

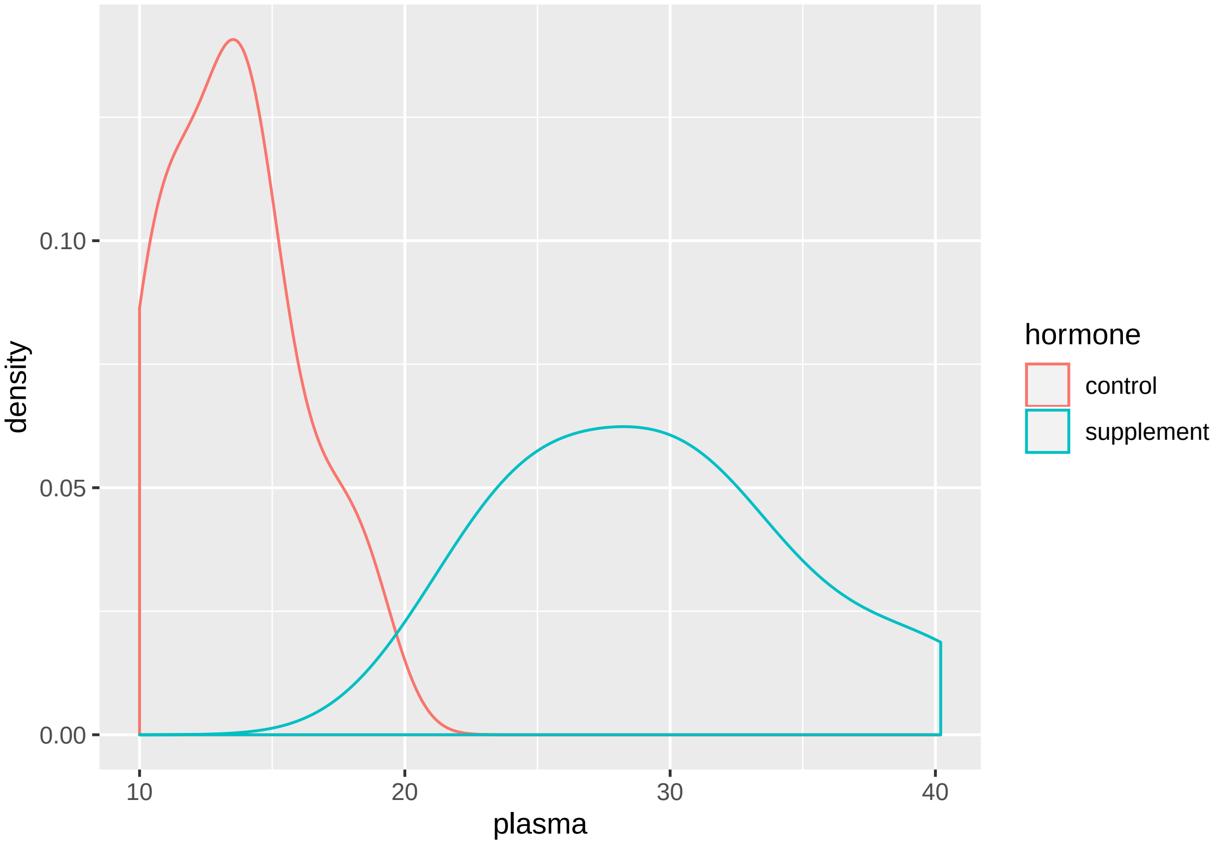
Problem 1: Bird Calcium

For many animals, the body's ability to use calcium depends on the level of certain sex-related hormones in the blood. The following data set looks at the relationship between hormone supplement (present or absent) and level of calcium in the blood. The subjects were 20 birds. Half the birds got a hormone supplement and the others served as controls. The response is the level of plasma calcium in mg/100 ml.

```
birds <- read.csv("http://www.evanlray.com/data/cobb_doe/bird_calcium_p160.csv") %>%
  transmute(
    hormone = ifelse(hormone == 1, "control", "supplement"),
    plasma = plasma
  )
```

(a) Check the conditions for conducting an analysis of these data with an ANOVA model. You should write an explicit sentence for each condition explaining why it is or isn't satisfied, with justification; if you need more information to make a determination, explain what else you would need to know. If necessary, find a transformation of the data so that the conditions are as well satisfied as possible.

```
ggplot(data = birds, mapping = aes(x = plasma, color = hormone)) +
  geom_density()
```



```
birds %>%
  group_by(hormone) %>%
  summarize(
    sd_plasma = sd(plasma)
  )
```

```
## # A tibble: 2 x 2
##   hormone      sd_plasma
##   <chr>            <dbl>
## 1 control    2.620856518
## 2 supplement 5.749792267
```

The conditions to check are:

- Independent observations: Without knowing more, I find it difficult to be sure our observations are independent. For example, we don't know how the birds were chosen to be in the study: maybe they were collected from two locations, and birds within each location are related and therefore have similar overall calcium levels in the blood. Similarly, since this is a study about a hormone supplement, it might be the case that female and male birds might respond to the hormone differently. In that case, the residuals for one male bird and another male bird might not be independent, in the sense that knowing one residual was positive (or negative) would give me information about whether another residual was likely to be positive or negative.

- Normally distributed errors: The distributions shown in the density plots are close enough to normally distributed for the t-based inferences to be approximately valid.

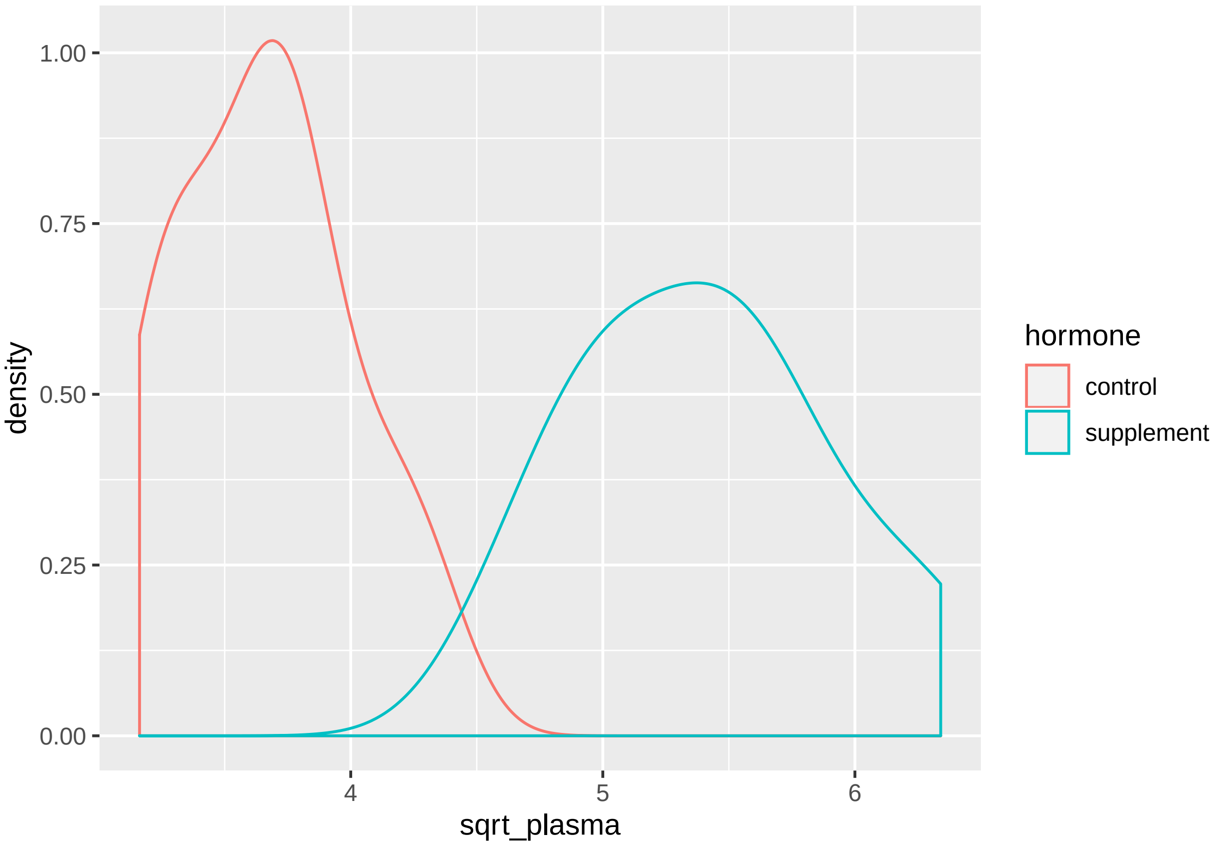
- Equal standard deviation in each group: This condition is not satisfied. The standard deviation is more than twice as large in the group that took the supplement than in the control group.

- No outliers: This condition is satisfied.

To make the standard deviations more similar in each group, we try a transformation. Since the group with the larger mean also has a larger standard deviation, we will try moving down on the ladder of powers. First we try a square root transformation:

```
birds <- birds %>%
  mutate(
    sqrt_plasma = sqrt(plasma)
  )

ggplot(data = birds, mapping = aes(x = sqrt_plasma, color = hormone)) +
  geom_density()
```



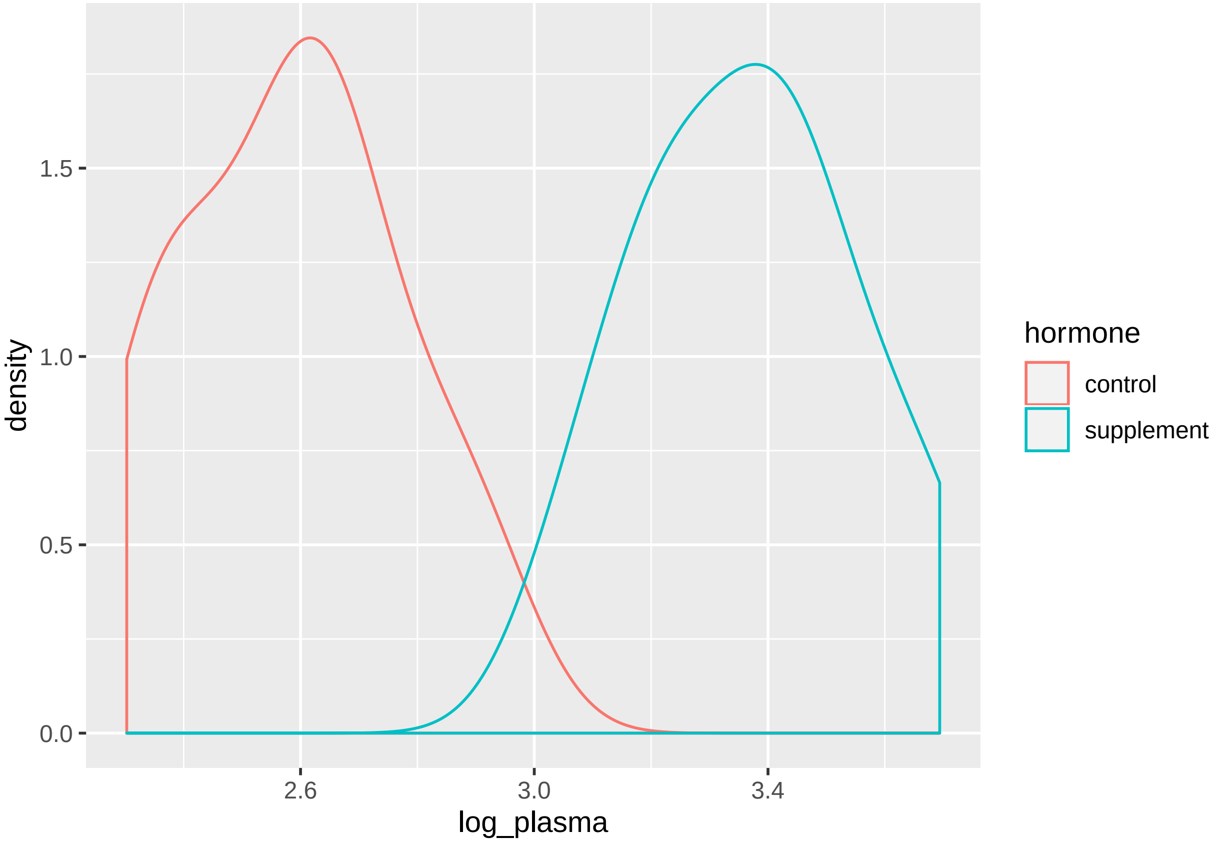
```
birds %>%
  group_by(hormone) %>%
  summarize(
    sd_plasma = sd(sqrt_plasma)
  )
```

```
## # A tibble: 2 x 2
##   hormone       sd_plasma
##   <chr>             <dbl>
## 1 control    0.3531632690
## 2 supplement 0.5249634181
```

That’s better - maybe even good enough. But there is still a difference in the standard deviations, so let’s continue exploring to see if we can do even better. The next step down is a log transformation.

```
birds <- birds %>%
  mutate(
    log_plasma = log(plasma)
  )

ggplot(data = birds, mapping = aes(x = log_plasma, color = hormone)) +
  geom_density()
```



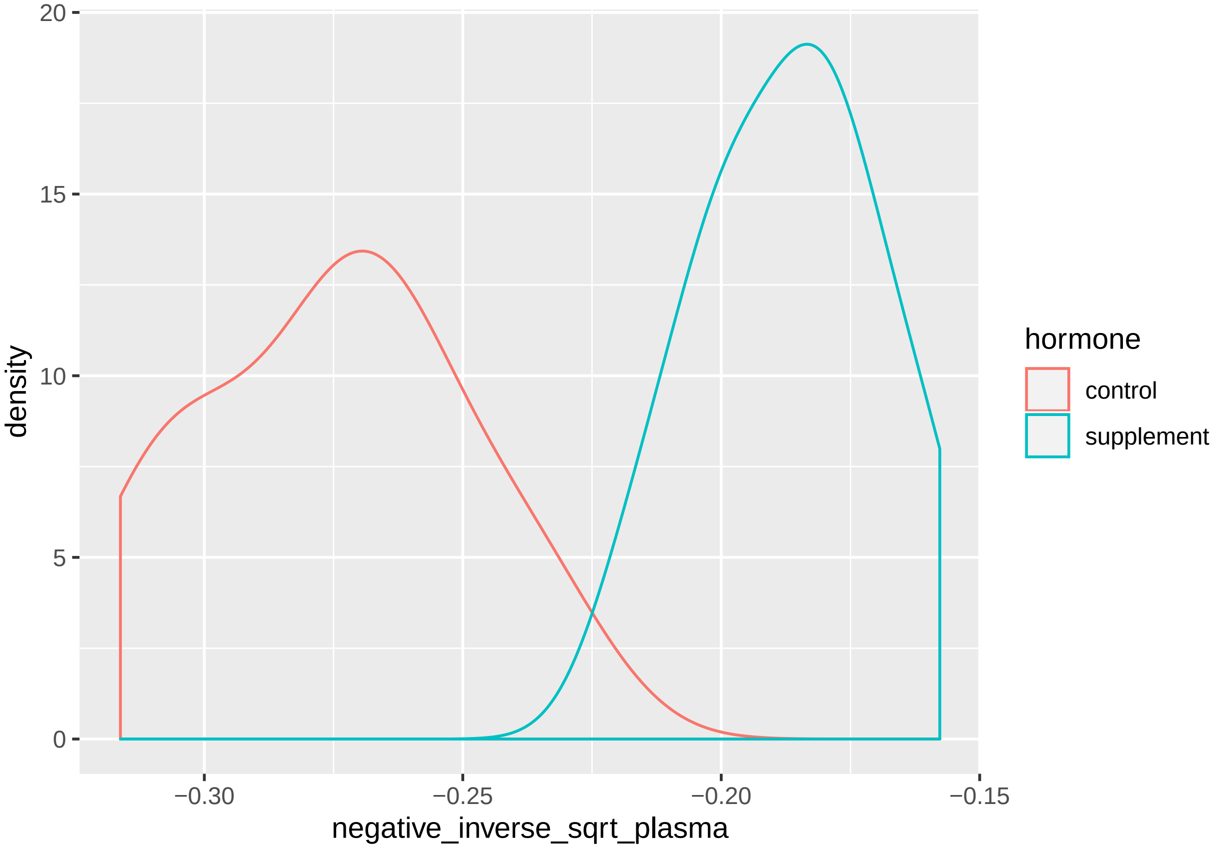
```
birds %>%
  group_by(hormone) %>%
  summarize(
    sd_plasma = sd(log_plasma)
  )
```

```
## # A tibble: 2 x 2
##   hormone       sd_plasma
##   <chr>             <dbl>
## 1 control    0.1919166671
## 2 supplement 0.1933674427
```

With the log transformation, the standard deviations are almost equal. This will probably be the transformation we use, but for the sake of completeness let's try going down one more step on the ladder. The next step on the ladder is $-y^{-0.5}$

```
birds <- birds %>%
  mutate(
    negative_inverse_sqrt_plasma = -1/sqrt(plasma)
  )

ggplot(data = birds, mapping = aes(x = negative_inverse_sqrt_plasma, color = hormone)) +
  geom_density()
```



```
birds %>%
  group_by(hormone) %>%
  summarize(
    sd_plasma = sd(negative_inverse_sqrt_plasma)
  )
```

```
## # A tibble: 2 x 2
##   hormone        sd_plasma
##   <chr>              <dbl>
## 1 control    0.02628381583
## 2 supplement 0.01795963899
```

Indeed, with this transformation the standard deviations for the two groups are more different than they were when we used the log transformation. We should stick with the log transformation. Based on the log transformation, all conditions are fairly well satisfied (other than perhaps the condition of independence, where we might like more information about whether or not any birds in the sample are related).

(b) For the purpose of this problem, let's assume that the conditions you checked in part (b) were fairly well satisfied (perhaps after suitable transformation). Conduct a test to find out whether there were any differences in the mean level of plasma calcium for birds taking the hormones and the control group (perhaps after suitable transformation). Please define all parameters involved, state your hypotheses in terms of equations involving the parameters and written sentences explaining what the hypotheses mean in context, and interpret the p-value for your test in terms of strength of evidence against the null hypothesis of the test, stated in context.

Define the parameters

μ_1 = mean of log plasma calcium in the population of birds not taking a hormone supplement.

μ_2 = mean of log plasma calcium in the population of birds taking a hormone supplement.

Our hypotheses are:

$H_0 : \mu_1 = \mu_2$. The average log plasma calcium is the same in the population of birds taking hormone supplements and not taking hormone supplements.

$H_A : \mu_1 \neq \mu_2$. The average log plasma calcium is different in the population of birds taking a hormone supplement and not taking a hormone supplement.

Note that the null hypothesis can be equivalently stated as $H_0 : \mu_2 - \mu_1 = 0$ or $H_0 : \mu_1 - \mu_2 = 0$. For the purpose of our test code, I will use the first of these.

```
model_fit <- lm(log_plasma ~ hormone, data = birds)
fit.contrast(model_fit, "hormone", c(-1, 1))
```

```
##                      Estimate Std. Error t value     Pr(>|t|)
## hormone c=( -1 1 ) 0.7790897 0.08615276 9.04312 4.101525e-08
## attr(,"class")
## [1] "fit_contrast"
```

```
fit.contrast(model_fit, "hormone", c(1, -1))
```

```
##                       Estimate Std. Error  t value     Pr(>|t|)
## hormone c=( 1 -1 ) -0.7790897 0.08615276 -9.04312 4.101525e-08
## attr(,"class")
## [1] "fit_contrast"
```

The p-value for the test is about 4.1×10^{-8}. The data provide extremely strong evidence against the null hypothesis of no difference between the group mean plasma counts on the log scale. It appears that the hormone supplement does have an association with calcium levels in the birds' blood.

(c) Find a 95% confidence interval describing the difference in the centers of the distributions of calcium concentrations between birds without the hormone supplement and birds with the hormone supplement. Interpret your confidence interval in context on the original (untransformed) scale of the data.

We can use essentially the same code as from part (b), but now need to obtain a confidence interval for the difference in means.

```
model_fit <- lm(log_plasma ~ hormone, data = birds)
fit.contrast(model_fit, "hormone", c(-1, 1), conf.int = 0.95)
```

```
##                      Estimate Std. Error t value     Pr(>|t|)  lower CI
## hormone c=( -1 1 ) 0.7790897 0.08615276 9.04312 4.101525e-08 0.5980895
##                    upper CI
## hormone c=( -1 1 )  0.96009
## attr(,"class")
## [1] "fit_contrast"
```

Since we performed inference after a log transformation, and the distributions of the transformed data were approximately symmetric, we can reverse the transformation in order to make statements about a multiplicative difference in medians on the original scale. We will need to exponentiate the confidence interval limits:

```
exp(0.598)
```

```
## [1] 1.818478
```

```
exp(0.960)
```

```
## [1] 2.611696
```

We are 95% confident that the median plasma calcium is between 1.8 and 2.6 times higher for birds taking a hormone supplement than for birds not taking a hormone supplement.

Problem 2: Pesticides in olive oil

Fenthion is a pesticide used against the olive fruit fly in olive groves. It is toxic to humans, so it is important that there be no residue left on the fruit or in olive oil that will be consumed. One theory was that, if there is residue of the pesticide left in the olive oil, it would dissipate over time. Chemists set out to test that theory by taking a random sample of small amounts of olive oil with fenthion residue and measuring the amount of fenthion in the oil at 3 different times over the year: day 0 (the day the sample was taken), day 281, and day 365.

The following R code reads in the data:

```
olives <- read_csv("http://www.evanlray.com/data/stat2/Olives.csv") %>%
  mutate(
    Day = factor(paste0("Day", Day))
  )
```

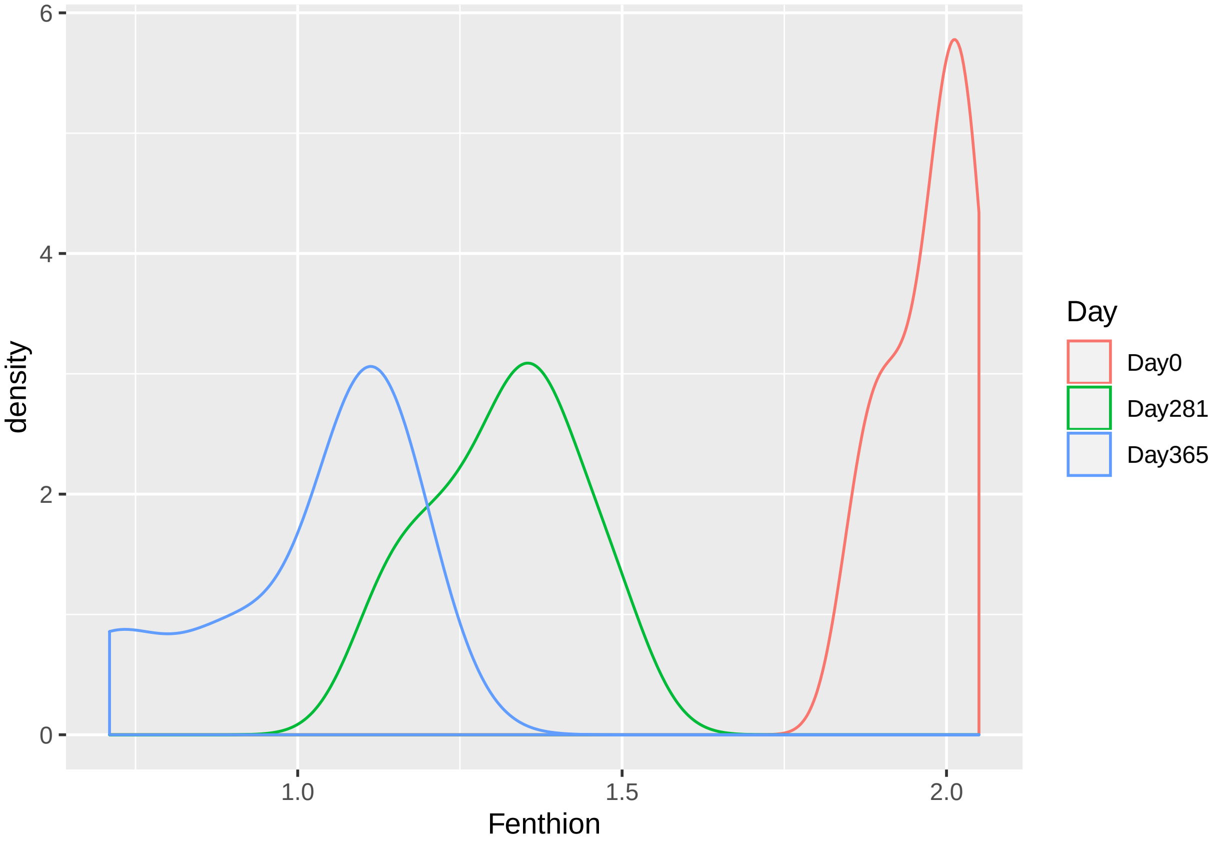
```
## Parsed with column specification:
## cols(
##   SampleNumber = col_double(),
##   Group = col_double(),
##   Day = col_double(),
##   Fenthion = col_double(),
##   FenthionSulphoxide = col_double(),
##   FenthionSulphone = col_double(),
##   Time = col_double()
## )
```

(a) Two variables in the model are `Fenthion` and `Day`; we will analyze these variables in this problem. Of these variables, which is the explanatory variable and which is the response? Explain.

Day is the explanatory variable and Fenthion is the response. We believe that the amount of fenthion in the oil may change over time, or that the day may explain variation in the amount of fenthion in the oil.

(b) Check the conditions for conducting an analysis of these data with an ANOVA model. You should write an explicit sentence for each condition explaining why it is or isn't satisfied, with justification; if you need more information to make a determination, explain what else you would need to know. If necessary, find a transformation of the data so that the conditions are as well satisfied as possible.

```
ggplot(data = olives, mapping = aes(x = Fenthion, color = Day)) +
  geom_density()
```



```
olives %>%
  group_by(Day) %>%
  summarize(
    sd_fenthion = sd(Fenthion)
  )
```

```
## # A tibble: 3 x 2
##   Day      sd_fenthion
##   <fct>          <dbl>
## 1 Day0   0.06774953874
## 2 Day281 0.1205680997
## 3 Day365 0.1726750320
```

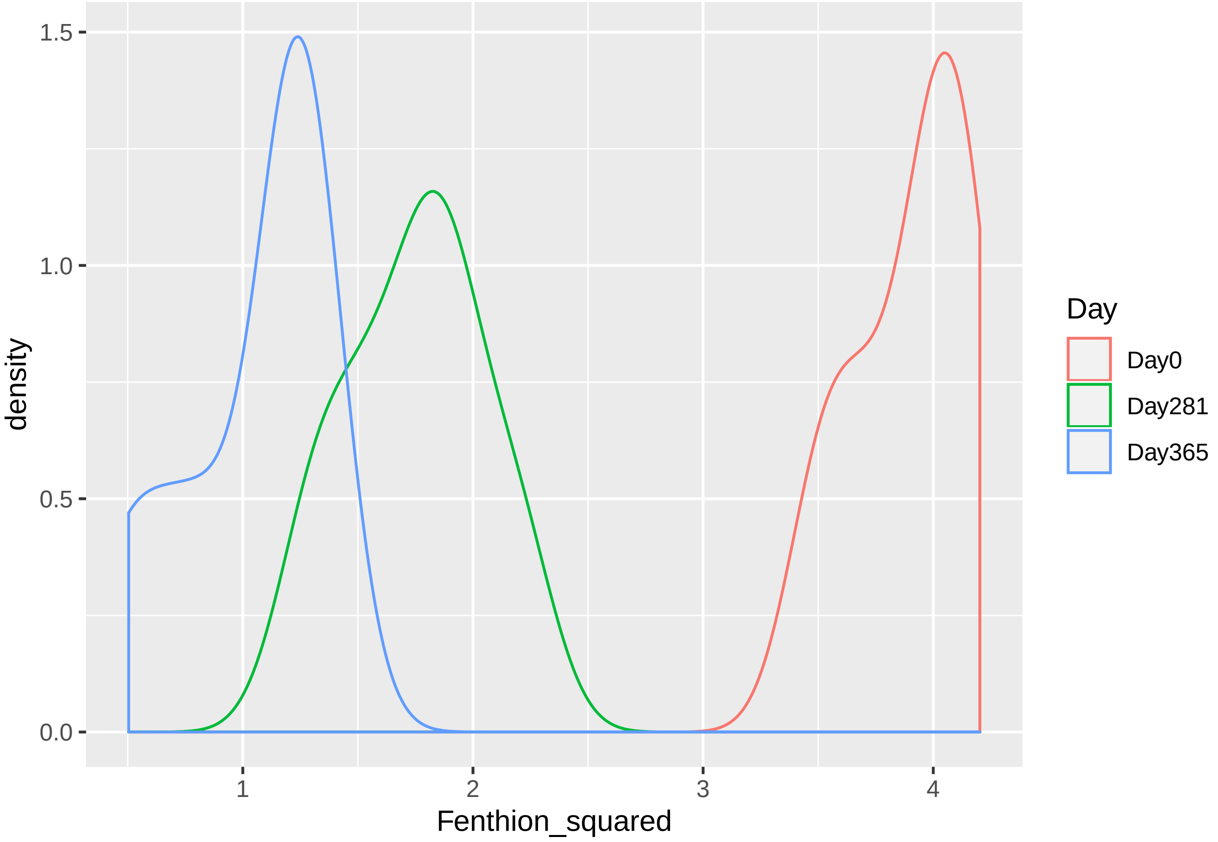
The conditions to check are:

- Independent observations: The observations are not independent. The researchers took a few samples of oil and measured the amount of fenthion in those samples over time. Two different measurements on the same oil sample would likely not be independent: if that oil had a very high amount of fenthion on day 0, it may still have a relatively high amount of oil on day 281 and day 365.
- Normally distributed errors: The distributions shown in the density plots are skewed to the left a little bith, but are close enough to normally distributed for the t-based inferences to be approximately valid.
- Equal standard deviation in each group: This condition is not satisfied. The standard deviation is nearly 3 times as large at day 365 than at day 0.
- No outliers: This condition is satisfied.

To make the standard deviations more similar in each group, we try a transformation. Since the group with the smaller mean has a larger standard deviation, we will try moving up on the ladder of powers. First we try a square transformation:

```
olives <- olives %>%
  mutate(
    Fenthion_squared = Fenthion^2
  )

ggplot(data = olives, mapping = aes(x = Fenthion_squared, color = Day)) +
  geom_density()
```



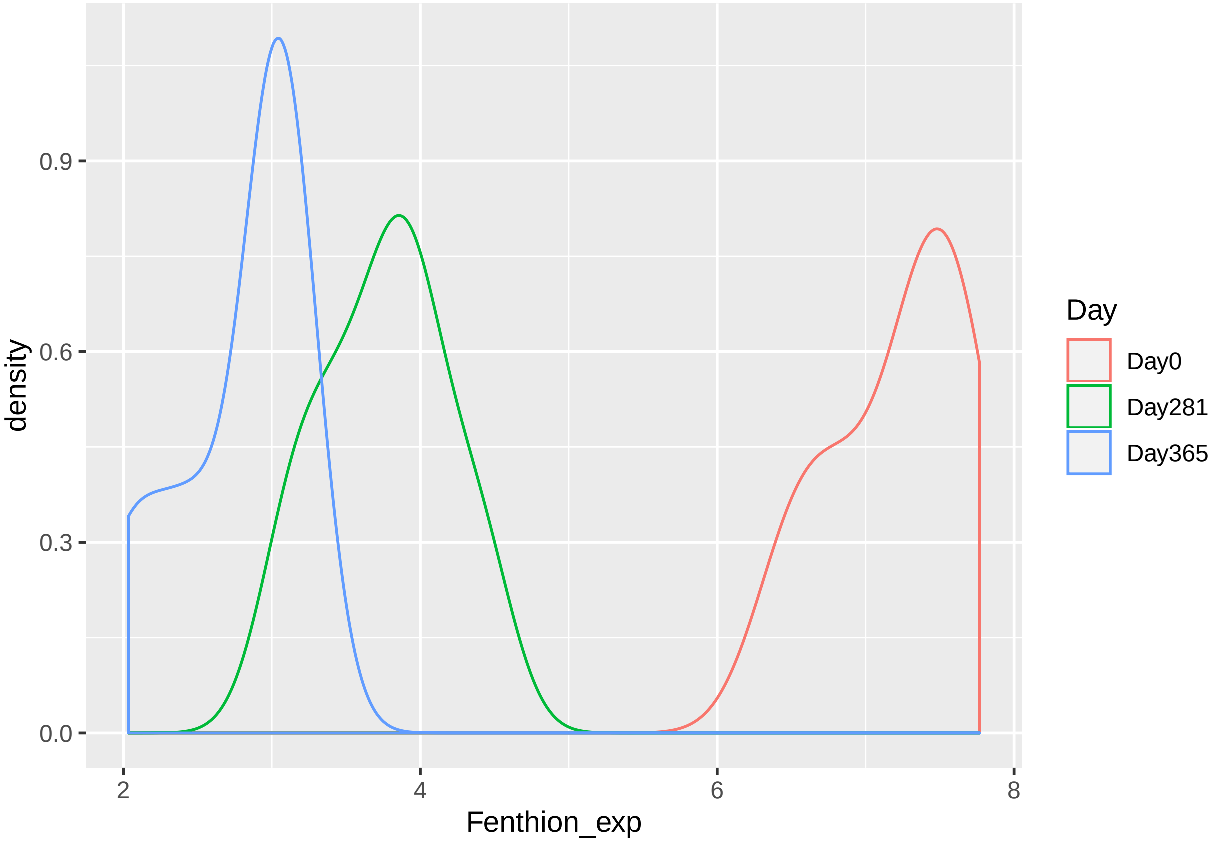
```
olives %>%
  group_by(Day) %>%
  summarize(
    sd_fenthion_squared = sd(Fenthion_squared)
  )
```

```
## # A tibble: 3 x 2
##   Day    sd_fenthion_squared
##   <fct>                <dbl>
## 1 Day0          0.2653640273
## 2 Day281        0.3153596132
## 3 Day365        0.3226386921
```

The standard deviations are much closer to being equal after the squaring transformation. These standard deviations are close enough that I'd be comfortable using this transformation, but let's keep exploring and see if we can do even better. The next step up on the ladder is either an exponential transformation or a cubic transformation, depending on the order of magnitude of the data; let's try an exponential transformation next.

```
olives <- olives %>%
  mutate(
    Fenthion_exp = exp(Fenthion)
  )

ggplot(data = olives, mapping = aes(x = Fenthion_exp, color = Day)) +
  geom_density()
```



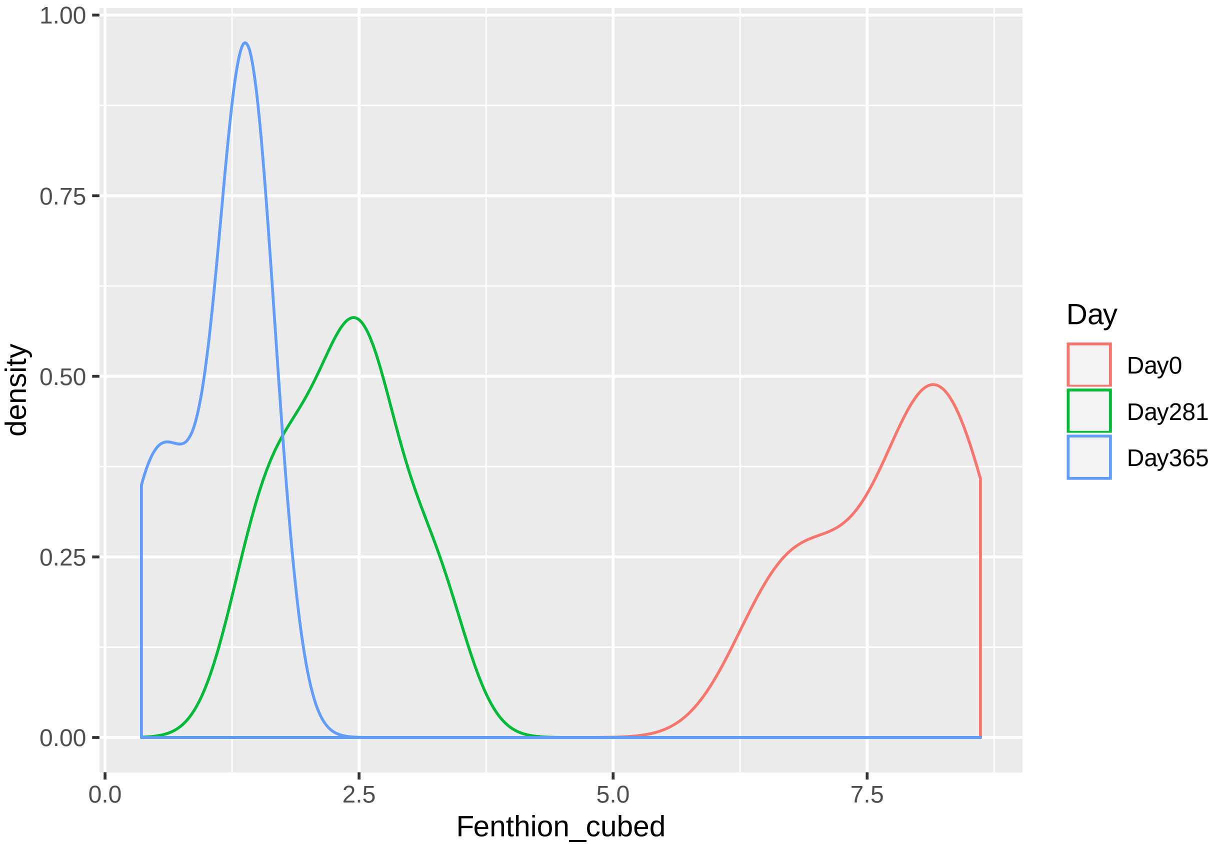
```
olives %>%
  group_by(Day) %>%
  summarize(
    sd_fenthion_exp = sd(Fenthion_exp)
  )
```

```
## # A tibble: 3 x 2
##   Day     sd_fenthion_exp
##   <fct>             <dbl>
## 1 Day0      0.4808057544
## 2 Day281    0.4478307349
## 3 Day365    0.4425330832
```

After an exponential transformation the standard deviations for the three groups are slightly closer to being equal than they were for the squared transformation. This transformation is slightly preferable. Before making a final decision, let's try one more step up on the ladder, a cubic transformation:

```
olives <- olives %>%
  mutate(
    Fenthion_cubed = Fenthion^3
  )

ggplot(data = olives, mapping = aes(x = Fenthion_cubed, color = Day)) +
  geom_density()
```



```
olives %>%
  group_by(Day) %>%
  summarize(
    sd_fenthion_cubed = sd(Fenthion_cubed)
  )
```

```
## # A tibble: 3 x 2
##   Day    sd_fenthion_cubed
##   <fct>              <dbl>
## 1 Day0        0.7801013605
## 2 Day281      0.6226429449
## 3 Day365      0.4607894387
```

This transformation is definitely worse than the first two transformations we tried. Let's go with the exponential transformation, which was slightly better than the squared transformation.

After this transformation the last three conditions are satisfied, but I do still have some serious concerns about whether or not the observations are independent.

(c) For the purpose of this problem, let's assume that the conditions you checked in part (b) were fairly well satisfied (perhaps after suitable transformation). Conduct a test to find out whether there were any differences in the mean amount of fenthion at the three different times

of year (if necessary, conduct a test about means on the transformed scale). Please define all parameters involved, state your hypotheses in terms of equations involving the parameters and written sentences explaining what the hypotheses mean in context, and interpret the p-value for your test in terms of strength of evidence against the null hypothesis of the test, stated in context.

Define the parameters

μ_1 = mean of exponentiated fenthiol levels in the population of oils at "Day 0" (just been processed?)

μ_2 = mean of exponentiated fenthiol levels in the population of oils at "Day 281" (281 days after being processed?)

μ_3 = mean of exponentiated fenthiol levels in the population of oils at "Day 365" (365 days after being processed?)

Our hypotheses are:

$H_0 : \mu_1 = \mu_2 = \mu_3$. The average exponentiated fenthiol levels are the same at all three days.

H_A : At least one of μ_1, μ_2, and μ_3 is not equal to the others. The average exponentiated fenthiol levels are different at different lengths of time after the oil was collected.

We will conduct an F test:

```r
model_fit <- lm(Fenthion_exp ~ Day, data = olives)
anova(model_fit)
```

```
## Analysis of Variance Table
##
## Response: Fenthion_exp
##           Df Sum Sq Mean Sq F value    Pr(>F)
## Day        2 65.244  32.622  155.95 9.176e-11 ***
## Residuals 15  3.138   0.209
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value for the test is about 9.2×10^{-11}. The data provide extremely strong evidence against the null hypothesis of no difference between the group mean fenthion levels on the exponentiated scale. It appears that the amount of fenthion in the oil does change over time.

(d) Find three confidence intervals with a familywise confidence level of 95%: one for the difference between the mean amount of fenthion present at day 0 and the mean amount present at day 281; a second for the difference between the mean amount of fenthion present at day 0 and the mean amount present at day 365; and a third for the difference between the mean amount of fenthion present at day 281 and the mean amount present at day 365. Find the confidence intervals using the Bonferroni adjustment for the familywise confidence level. Interpret your confidence intervals in context. For which pairs of days do the data provide statistically significant evidence of a difference in means? All of your inferences can be on the transformed scale, if you selected a transformation in part (b).

Since we are finding three confidence intervals, in the Bonferroni calculations the value of k is 3. The quantile of the t distribution to use for the multiplier is therefore $1 - \frac{0.05}{2\times3} \approx 0.9916667$. Each individual confidence interval will be an approximate $(1 - 0.05/3) \times 100\% = 98.3\%$ confidence interval. Our sample size is 18, and there are 3 groups, so the degrees of freedom is 18 - 3 = 15. The multiplier is therefore

```r
qt(0.9916667, df = 18 - 3)
```

```
## [1] 2.693741
```

The code below finds these intervals “by hand” using the formula with the Bonferroni multiplier. Here, the `fit.contrast` function is used only to find the estimate and its standard error.

```r
# first comparison: day 0 vs day 281.
fit.contrast(model_fit, "Day", c(1, -1, 0), conf.int = 0.95)
```

```
##                     Estimate Std. Error  t value     Pr(>|t|) lower CI
## Day c=( 1 -1 0 ) 3.460563  0.2640627 13.10508 1.285247e-09 2.897727
##                     upper CI
## Day c=( 1 -1 0 )   4.0234
## attr(,"class")
## [1] "fit_contrast"
```

```r
3.460563 - 2.693741 * 0.2640627
```

```
## [1] 2.749246
```

```r
3.460563 + 2.693741 * 0.2640627
```

```
## [1] 4.17188
```

```r
# second comparison: day 0 vs day 365
fit.contrast(model_fit, "Day", c(1, 0, -1), conf.int = 0.95)
```

```
##                     Estimate Std. Error  t value     Pr(>|t|) lower CI
## Day c=( 1 0 -1 ) 4.437542  0.2640627 16.80488 3.864877e-11 3.874706
##                     upper CI
## Day c=( 1 0 -1 ) 5.000378
## attr(,"class")
## [1] "fit_contrast"
```

```r
4.437542 - 2.693741 * 0.2640627
```

```
## [1] 3.726225
```

```r
4.437542 + 2.693741 * 0.2640627
```

```
## [1] 5.148859
```

```r
# third comparison: day 281 vs day 365
fit.contrast(model_fit, "Day", c(0, 1, -1), conf.int = 0.95)
```

```
##                      Estimate Std. Error  t value    Pr(>|t|)  lower CI
## Day c=( 0 1 -1 ) 0.9769785  0.2640627 3.699797 0.002139976 0.4141422
##                     upper CI
## Day c=( 0 1 -1 ) 1.539815
## attr(,"class")
## [1] "fit_contrast"
```

```r
0.9769785 - 2.693741 * 0.2640627
```

```
## [1] 0.265662
```

```r
0.9769785 + 2.693741 * 0.2640627
```

```
## [1] 1.688295
```

Really all we did here was find three individual 98.3% confidence intervals so that the familywise confidence level was 95%. We could just do this calculation directly using `fit.contrast`:

```r
fit.contrast(model_fit, "Day", c(1, -1, 0), conf.int = 0.983333)
```

```
##                     Estimate Std. Error  t value     Pr(>|t|) lower CI
## Day c=( 1 -1 0 ) 3.460563  0.2640627 13.10508 1.285247e-09  2.74925
##                   upper CI
## Day c=( 1 -1 0 ) 4.171877
## attr(,"class")
## [1] "fit_contrast"
```

```
# second comparison: day 0 vs day 365
fit.contrast(model_fit, "Day", c(1, 0, -1), conf.int = 0.9833)
```

```
##                   Estimate Std. Error  t value     Pr(>|t|) lower CI
## Day c=( 1 0 -1 ) 4.437542  0.2640627 16.80488 3.864877e-11 3.726489
##                   upper CI
## Day c=( 1 0 -1 ) 5.148595
## attr(,"class")
## [1] "fit_contrast"
```

```
# third comparison: day 281 vs day 365
fit.contrast(model_fit, "Day", c(0, 1, -1), conf.int = 0.9833)
```

```
##                    Estimate Std. Error  t value    Pr(>|t|)  lower CI
## Day c=( 0 1 -1 ) 0.9769785  0.2640627 3.699797 0.002139976 0.2659257
##                   upper CI
## Day c=( 0 1 -1 ) 1.688031
## attr(,"class")
## [1] "fit_contrast"
```

Either way the calculations are done, we arrive at the same three confidence intervals.

We are 95% confident that the difference in means of exponentiated fenthiol levels in the population of oils at Day 0 and at Day 281 is between 2.749 and 4.172, the difference in means of exponentiated fenthiol levels in the population of oils at Day 0 and Day 365 is between 3.726 and 5.149, and the difference in means of exponentiated fenthiol levels in the population of oils at Day 281 and Day 365 is between 0.266 and 1.688. For 95% of samples, a set of three confidence intervals calculated in this way would *simultaneously* contain the respective differences in means they are estimating.