Lab06 - Transformations for ANOVA

Goals

The goal in this lab is to practice working with transformations for ANOVA.

Loading packages

Here are some packages with functionality you may need for this lab. Run this code chunk now.

```
library(readr)
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.5.2
```

```
library(gridExtra)
library(mosaic)
```

```
## Warning: package 'mosaic' was built under R version 3.5.2

## Loading required package: dplyr

## Warning: package 'dplyr' was built under R version 3.5.2

##
## Attaching package: 'dplyr'

## The following object is masked from 'package:gridExtra':
##
##     combine

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

## Loading required package: lattice

## Loading required package: ggformula

## Warning: package 'ggformula' was built under R version 3.5.2

## Loading required package: ggstance

##
## Attaching package: 'ggstance'

## The following objects are masked from 'package:ggplot2':
##
##     geom_errorbarh, GeomErrorbarh

##
## New to ggformula?  Try the tutorials:
##  learnr::run_tutorial("introduction", package = "ggformula")
##  learnr::run_tutorial("refining", package = "ggformula")

## Loading required package: mosaicData
```

```
## Loading required package: Matrix
```

```
##
## The 'mosaic' package masks several functions from core packages in order to add
## additional features.  The original behavior of these functions should not be affected by this.
##
## Note: If you use the Matrix package, be sure to load it BEFORE loading mosaic.
```

```
##
## Attaching package: 'mosaic'
```

```
## The following object is masked from 'package:Matrix':
##
##     mean
```

```
## The following objects are masked from 'package:dplyr':
##
##     count, do, tally
```

```
## The following object is masked from 'package:ggplot2':
##
##     stat
```

```
## The following objects are masked from 'package:stats':
##
##     binom.test, cor, cor.test, cov, fivenum, IQR, median,
##     prop.test, quantile, sd, t.test, var
```

```
## The following objects are masked from 'package:base':
##
##     max, mean, min, prod, range, sample, sum
```

```
library(dplyr)

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

A gas chromatograph is an instrument that measures the amounts of various compounds contained in a sample by separating the various constituents. The total number of counts recorded by the chromatograph is proportional to the amount of the compound present.

A calibration experiment was performed to see how the recorded counts from the chromatograph related to the concentration of a compound in a mixture and the flow rate through the chromatograph. In this lab we will just look at the relationship between the concentration (explanatory variable) and the counts (response variable).

```
chromatography <- read_csv("http://www.evanlray.com/data/sdm3/Chapter_29/Ch29_Chromatography.csv")
```

```
## Parsed with column specification:
## cols(
##   Concentration = col_character(),
##   `Flow Rate` = col_character(),
##   Counts = col_double()
## )
```

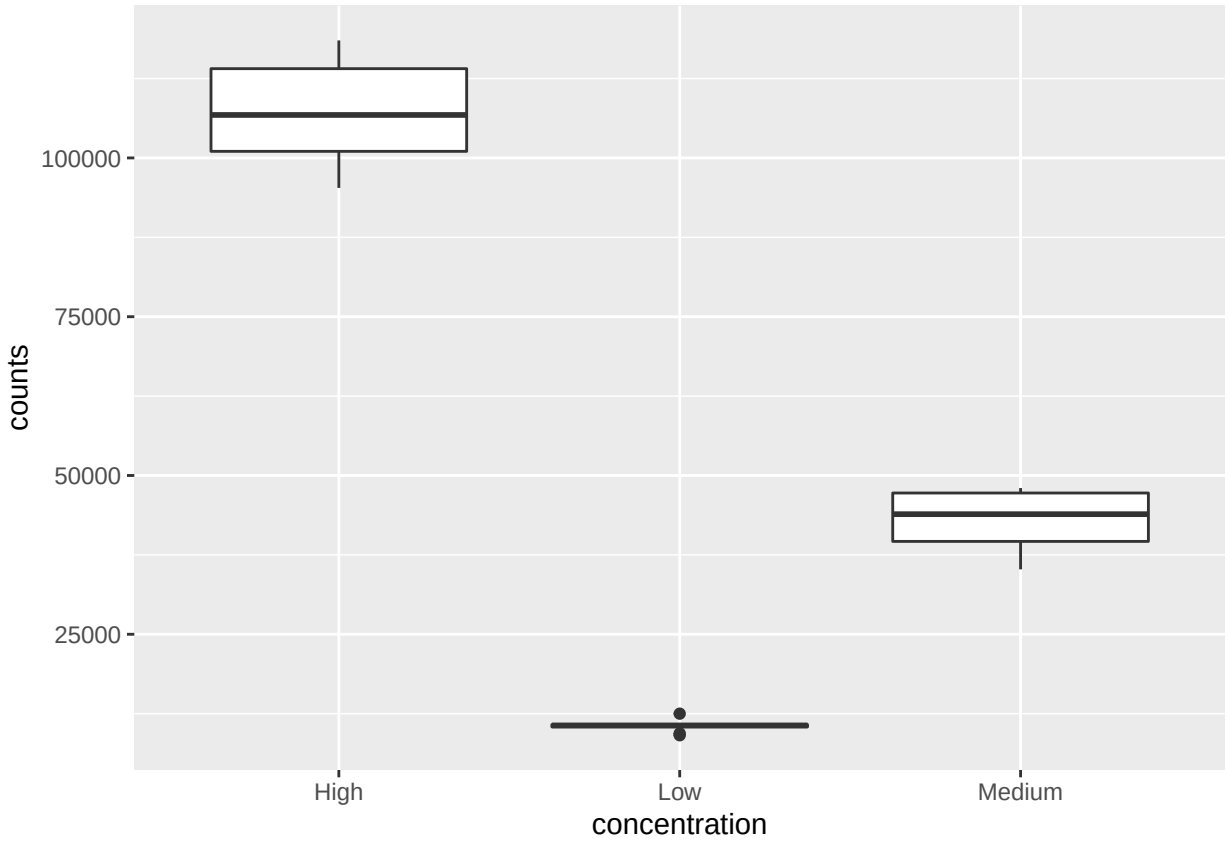
```
names(chromatography) <- c("concentration", "flow_rate", "counts")

chromatography %>%
  count(concentration)
```

```
## # A tibble: 3 x 2
```

```
##   concentration     n
##   <chr>         <int>
## 1 High             10
## 2 Low              10
## 3 Medium           10
```

1. Make an appropriate plot of the data: it might be nice to use a histogram or density plot, separately for each value of cylinders. Also calculate the standard deviation for each group. Would it be appropriate to use an ANOVA model for these data?

```
ggplot(data = chromatography, mapping = aes(x = concentration, y = counts)) +
  geom_boxplot()
```

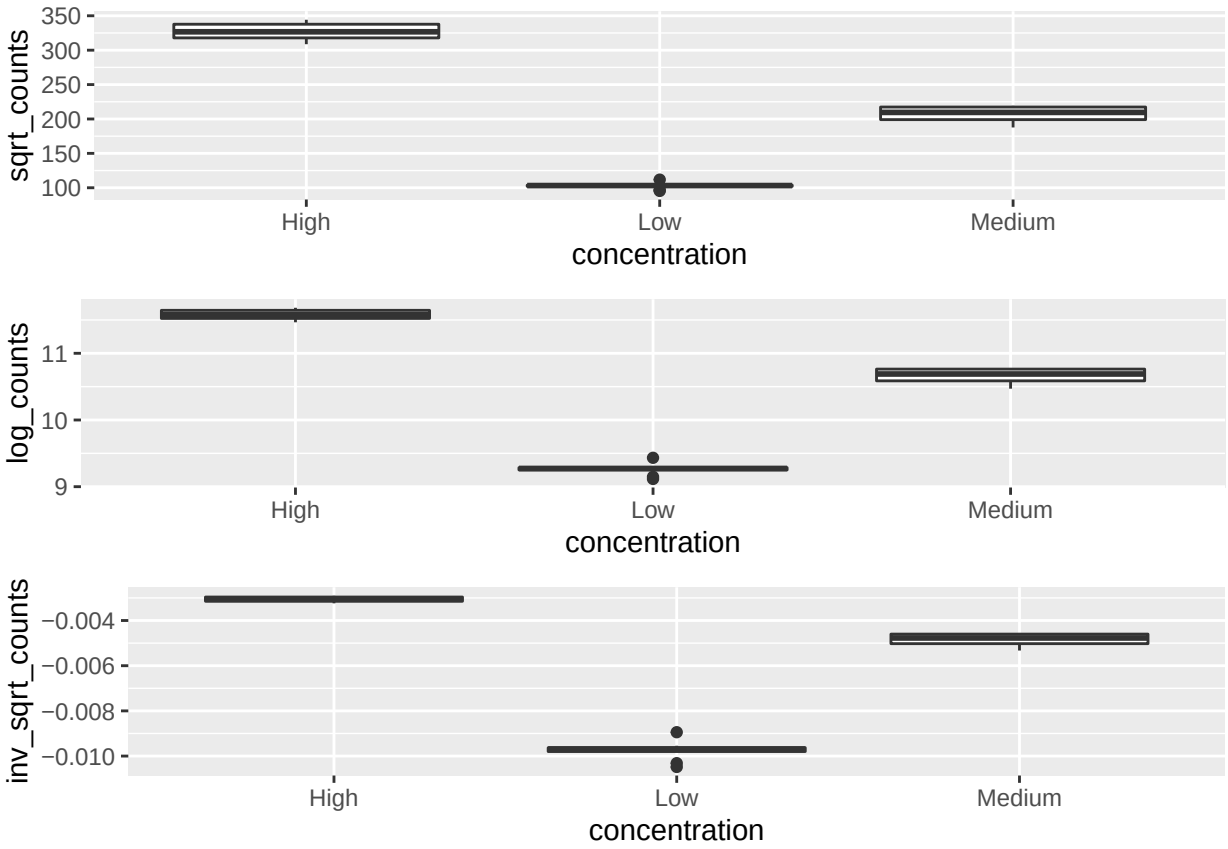



```
chromatography %>%
  group_by(concentration) %>%
  summarize(
    sd(counts)
  )
```

```
## # A tibble: 3 x 2
##   concentration `sd(counts)`
##   <chr>                <dbl>
## 1 High          8641.856796
## 2 Low            915.9718579
## 3 Medium        4497.556868
```

2. Find a transformation of the data so that the ANOVA model would be appropriate.

```
chromatography <- chromatography %>%
  mutate(
    sqrt_counts = sqrt(counts),
    log_counts = log(counts),
    inv_sqrt_counts = -1/sqrt(counts)
  )

p_sqrt <- ggplot(data = chromatography, mapping = aes(x = concentration, y = sqrt_counts)) +
  geom_boxplot()
p_log <- ggplot(data = chromatography, mapping = aes(x = concentration, y = log_counts)) +
  geom_boxplot()
p_inv_sqrt <- ggplot(data = chromatography, mapping = aes(x = concentration, y = inv_sqrt_counts)) +
  geom_boxplot()

grid.arrange(p_sqrt, p_log, p_inv_sqrt)
```




```
chromatography %>%
  group_by(concentration) %>%
  summarize(
    sd(sqrt_counts),
    sd(log_counts),
    sd(inv_sqrt_counts)
  )
```

```
## # A tibble: 3 x 4
##   concentration `sd(sqrt_counts)` `sd(log_counts)` `sd(inv_sqrt_counts)`
##   <chr>                     <dbl>            <dbl>                  <dbl>
```

```
## 1 High                13.21413071     0.08090121936       0.0001239476822
## 2 Low                  4.434431352    0.08611873813       0.0004192992433
## 3 Medium              10.97834749     0.1074039081        0.0002632217346
```

The standard deviations in the different groups are most consistent with a log transformations. The distributions are close enough to normally distributed.

3. Conduct a test of the claim that the mean count is the same for all three concentration levels.

This is formally a test of the claim that the mean of the log count is the same for all three concentration levels, since we're working with log-transformed data.

$H_0 : \mu_1 = \mu_2 = \mu_3$ H_A : at least one of μ_1, μ_2, and μ_3 is not equal to the others

```
model_fit <- lm(log_counts ~ concentration, data = chromatography)
summary(model_fit)
```

```
##
## Call:
## lm(formula = log_counts ~ concentration, data = chromatography)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.19663 -0.05638  0.01090  0.06481  0.17162
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         11.57961    0.02915  397.20   <2e-16 ***
## concentrationLow    -2.31775    0.04123  -56.22   <2e-16 ***
## concentrationMedium -0.91361    0.04123  -22.16   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09219 on 27 degrees of freedom
## Multiple R-squared:  0.9917, Adjusted R-squared:  0.991
## F-statistic:  1604 on 2 and 27 DF,  p-value: < 2.2e-16
```

The p-value for the test is less than 2.2×10^{-16}. There is extremely strong evidence against the null hypothesis that the mean of the log count is the same for all three concentration levels.

4. Report and interpret an estimate of the difference in the centers of the distributions of counts for the high concentration and the low concentration, as well as a 95% confidence interval for that difference. You should be able to do this in a few different ways.

```
fit <- lm(log_counts ~ concentration, data = chromatography)
summary(fit)
```

```
##
## Call:
## lm(formula = log_counts ~ concentration, data = chromatography)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.19663 -0.05638  0.01090  0.06481  0.17162
##
```

```
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         11.57961    0.02915  397.20   <2e-16 ***
## concentrationLow    -2.31775    0.04123  -56.22   <2e-16 ***
## concentrationMedium -0.91361    0.04123  -22.16   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09219 on 27 degrees of freedom
## Multiple R-squared:  0.9917, Adjusted R-squared:  0.991
## F-statistic:  1604 on 2 and 27 DF,  p-value: < 2.2e-16
```

```
confint(fit)
```

```
##                         2.5 %     97.5 %
## (Intercept)         11.519790 11.6394240
## concentrationLow    -2.402342 -2.2331533
## concentrationMedium -0.998200 -0.8290115
```

```
exp(-2.31775)
```

```
## [1] 0.09849495
```

We estimate that the median count in the low group is about 0.1 times the median count for the high group.

We can verify this by looking at the actual medians for these groups:

```
chromatography %>%
  group_by(concentration) %>%
  summarize(
    median(counts)
  )
```

```
## # A tibble: 3 x 2
##   concentration `median(counts)`
##   <chr>                    <dbl>
## 1 High                    106765
## 2 Low                      10650
## 3 Medium                   43910
```

OR

```
exp(2.31775)
```

```
## [1] 10.1528
```

We estimate that the median count in the high group is about 10 times the median concentration for the low group.