Examples: Transformations for ANOVA models

Sleuth3 Sections 3.5 and 5.5

Example: Cloud Seeding (Sleuth3 Case Study 3.1.1)

Quote from book: "On each of 52 days that were deemed suitable for cloud seeding, a random mechanism was used to decide whether to seed the target cloud on that day or to leave it unseeded as a control. ... [P]recipitation was measured as the total rain volume falling from the cloud base following the airplane seeding run."

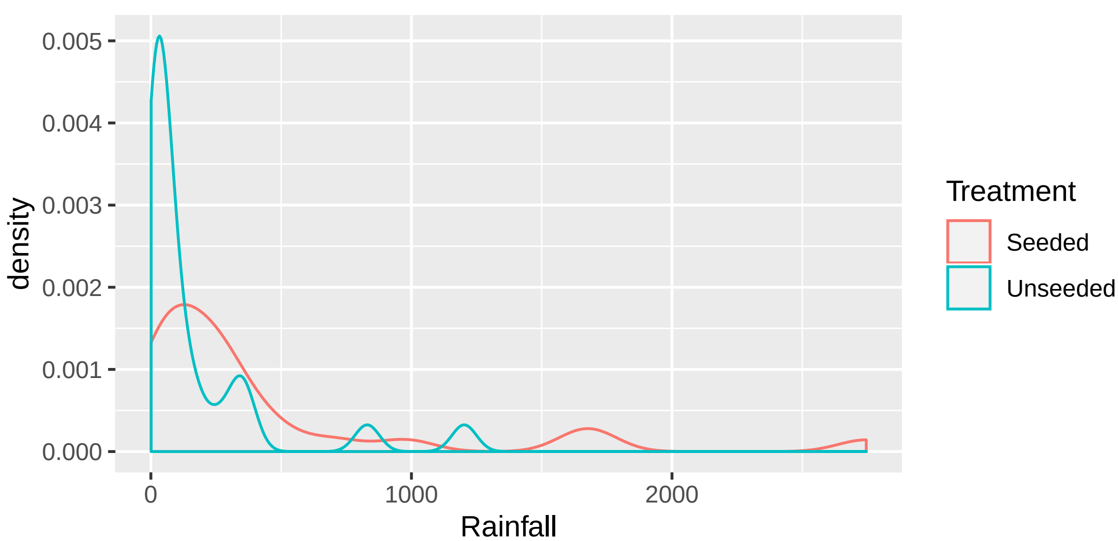
```
clouds <- read_csv("http://www.evanlray.com/data/sleuth3/case0301_cloud_seeding.csv")
head(clouds, 4)
```

```
## # A tibble: 4 x 2
##   Rainfall Treatment
##      <dbl> <chr>
## 1    1203. Unseeded
## 2     830. Unseeded
## 3     372. Unseeded
## 4     346. Unseeded
```

Starting Point

Here are density plots and box plots, separately for each Treatment.

```
ggplot(data = clouds, mapping = aes(x = Rainfall, color = Treatment)) +
  geom_density()
```



Standard deviations for each group:

```
clouds %>%
  group_by(Treatment) %>%
  summarize(
    sd_rainfall = sd(Rainfall)
  )
```

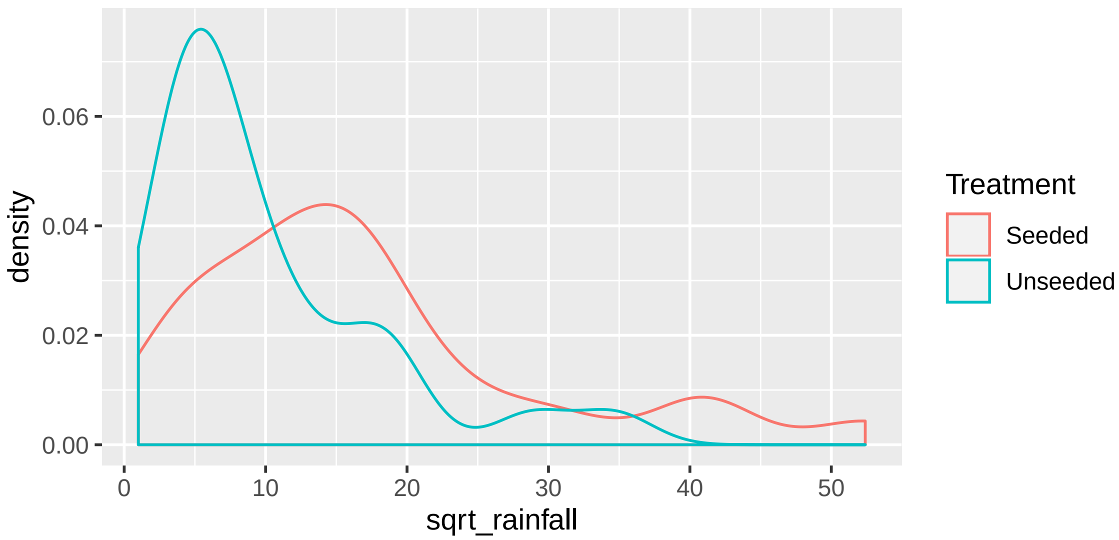
```
## # A tibble: 2 x 2
##   Treatment sd_rainfall
##   <chr>           <dbl>
## 1 Seeded           651.
## 2 Unseeded         278.
```

The standard deviations are very different, and the distributions are skewed right, so move down one step on the ladder.

Down 1 Step: $\sqrt{Rainfall}$

```
clouds <- clouds %>%
  mutate(
    sqrt_rainfall = sqrt(Rainfall)
  )
```

```
ggplot(data = clouds, mapping = aes(x = sqrt_rainfall, color = Treatment)) +
  geom_density()
```




```
clouds %>%
  group_by(Treatment) %>%
  summarize(
    sd_rainfall = sd(sqrt_rainfall)
  )
```

```
## # A tibble: 2 x 2
##   Treatment sd_rainfall
##   <chr>           <dbl>
## 1 Seeded          12.5
## 2 Unseeded         8.24
```

These distributions are closer to symmetric – probably good enough.

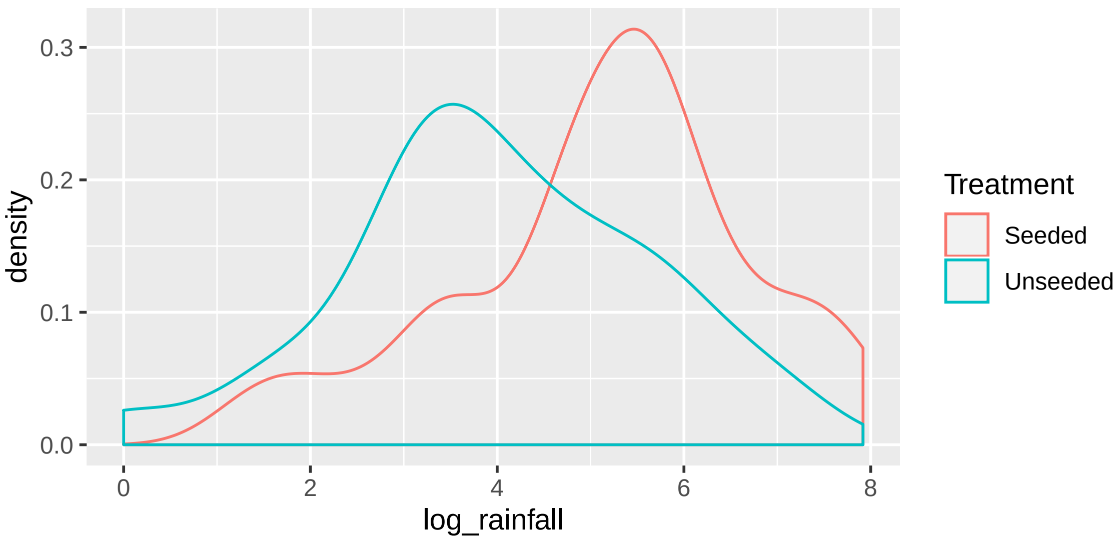
The ratio of these standard deviations is less than 2 – often used as a guide for when we're OK.

However, we can make it even better if we go down another step.

Down 2 Steps: $\log(Rainfall)$

```
clouds <- clouds %>%
  mutate(
    log_rainfall = log(Rainfall)
  )
```

```
ggplot(data = clouds, mapping = aes(x = log_rainfall, color = Treatment)) +
  geom_density()
```




```
clouds %>%
  group_by(Treatment) %>%
  summarize(
    sd_rainfall = sd(log_rainfall)
  )
```

```
## # A tibble: 2 x 2
##   Treatment sd_rainfall
##   <chr>           <dbl>
## 1 Seeded           1.60
## 2 Unseeded         1.64
```

Good enough! We can conduct our analysis on this scale.

Analysis on transformed scale

```
clouds %>%
  group_by(Treatment) %>%
  summarize(
    mean_log_rainfall = mean(log_rainfall)
  )
```

```
## # A tibble: 2 x 2
##   Treatment mean_log_rainfall
##   <chr>                 <dbl>
## 1 Seeded                 5.13
## 2 Unseeded               3.99
```

```
rainfall_fit <- lm(log_rainfall ~ Treatment, data = clouds)
summary(rainfall_fit)
```

```
##
## Call:
## lm(formula = log_rainfall ~ Treatment, data = clouds)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.9904 -0.7453  0.1624  1.0187  3.1018
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)         5.1342     0.3179  16.152   <2e-16 ***
## TreatmentUnseeded  -1.1438     0.4495  -2.544   0.0141 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.621 on 50 degrees of freedom
## Multiple R-squared:  0.1146, Adjusted R-squared:  0.09693
## F-statistic: 6.474 on 1 and 50 DF,  p-value: 0.01408
```

```
confint(rainfall_fit)
```

```
##                       2.5 %    97.5 %
## (Intercept)        4.495729  5.772645
## TreatmentUnseeded -2.046697 -0.240865
```

```
library(gmodels)
fit.contrast(rainfall_fit, "Treatment", c(1, -1), conf.int = 0.95)
```

```
##                        Estimate Std. Error  t value   Pr(>|t|) lower CI
## Treatment c=( 1 -1 ) 1.143781  0.4495342 2.544369 0.01408266 0.240865
##                        upper CI
## Treatment c=( 1 -1 ) 2.046697
## attr(,"class")
## [1] "fit_contrast"
```

We can interpret these numbers either on the new, transformed, data scale or on the original data scale.

1. Interpret the group mean estimates above on the transformed scale (always works!):

2. Interpret the group mean estimates above on the original data scale (works if we got to a place where distributions were approximately symmetric after transformation!):

```r
exp(5.13)
```

```
## [1] 169.0171
```

```r
exp(3.99)
```

```
## [1] 54.05489
```

3. Interpret the estimated difference in means above on the transformed scale (always works!):

4. Interpret the estimted difference in means above on the original data scale (works only if the transformation selected was the log transformation and the resulting distribution was approximately symmetric!):

```r
exp(1.143781)
```

```
## [1] 3.138613
```

```r
exp(0.240865)
```

```
## [1] 1.272349
```

```r
exp(2.046697)
```

```
## [1] 7.742286
```