ANOVA: Transformations

Sleuth3 Sections 3.5 and 5.5

Context

- Transformations can sometimes help with the following issues:
  - non-normal distributions within each group (but skewness is only a problem if it is very serious)
  - lack of equal variance for all groups
  - outliers (but usually only if this is a side effect of serious skewness)
- The most common transformations (that we'll consider in this class) work for positive values only.

The Ladder of Powers

- Imagine a "ladder of powers" of y: We start at y and go up or down the ladder.

Transformation	Comments
$\vdots$	
e^y	Exactly where on the ladder the exponential transformation belongs depends on the magnitude of the data, but somewhere around here...
y^2	
y	Start here (no transformation)
$\sqrt{y}$	
$y^{\text{“}0\text{”}}$	We use $\log(y)$ here
$-1/\sqrt{y}$	The $-$ keeps the values of y in order
$-1/y$	
$-1/y^2$	
$\vdots$	

- Which direction?
  - If a variable is skewed right, move it down the ladder (pull down large values)
  - If a variable is skewed left, move it up the ladder (pull up small values)

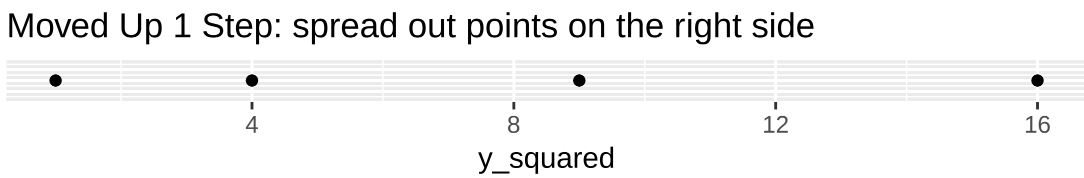
Some (minimal) facts about logarithms and exponentials

- Foundations:
  - In this class the base of our logarithms is e
  - Notation: $\exp(x) = e^x$
- log() and exp() are inverses
  - $\log(\exp(x)) = x$
  - $\exp(\log(x)) = x$
- They are useful because they convert multiplication to addition, and addition to multiplication
  - $\log(a \cdot b) = \log(a) + \log(b)$
  - $\exp(a + b) = \exp(a) \cdot \exp(b)$

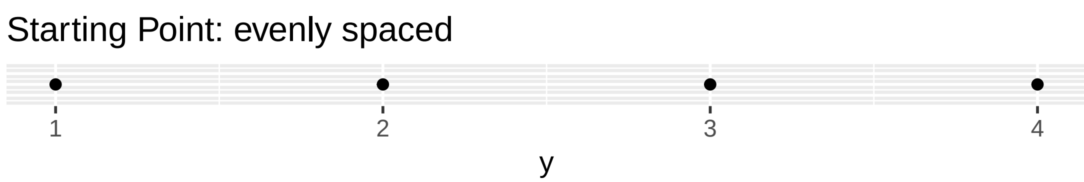
```r
example <- data.frame(
  y = c(1, 2, 3, 4),
    y_squared = c(1, 2, 3, 4)^2,
    sqrt_y = c(1, 2, 3, 4)^0.5
)
example
```

```
##   y y_squared   sqrt_y
## 1 1         1 1.000000
## 2 2         4 1.414214
## 3 3         9 1.732051
## 4 4        16 2.000000
```

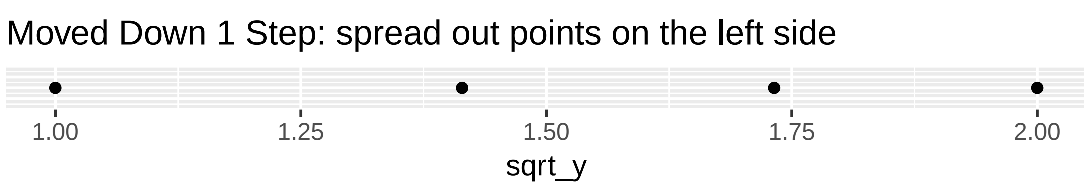
```r
ggplot(data = example, mapping = aes(x = y_squared, y = 0)) +
  geom_point() +
  ggtitle("Moved Up 1 Step: spread out points on the right side") +
  ylab("") +
  theme(axis.text.y=element_blank(), axis.ticks.y = element_blank())
```




```r
ggplot(data = example, mapping = aes(x = y, y = 0)) +
  geom_point() +
  ggtitle("Starting Point: evenly spaced") +
  ylab("") +
  theme(axis.text.y=element_blank(), axis.ticks.y = element_blank())
```




```r
ggplot(data = example, mapping = aes(x = sqrt_y, y = 0)) +
  geom_point() +
  ggtitle("Moved Down 1 Step: spread out points on the left side") +
  ylab("") +
  theme(axis.text.y=element_blank(), axis.ticks.y = element_blank())
```



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. An airplane flew through the cloud in both cases. . . . [p]recipitation was measured as the total rain volume falling from the cloud base following the airplane seeding run."

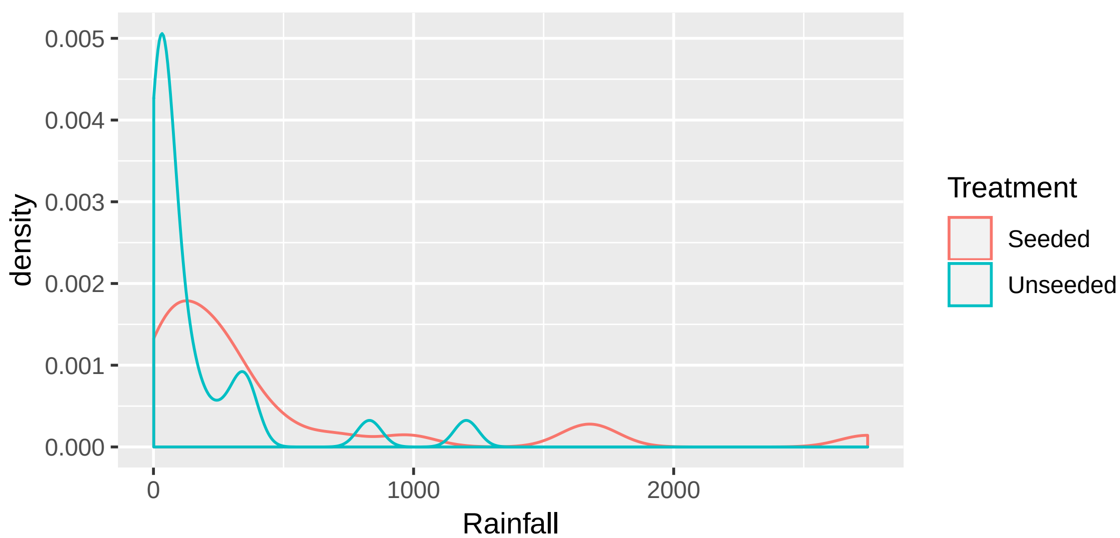
```
clouds <- read_csv("http://www.evanlray.com/data/sleuth3/case0301_cloud_seeding.csv")
head(clouds)
```

```
## # A tibble: 6 x 2
##   Rainfall Treatment
##      <dbl> <chr>
## 1    1203. Unseeded
## 2     830. Unseeded
## 3     372. Unseeded
## 4     346. Unseeded
## 5     321. Unseeded
## 6     244. 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)
  )
```

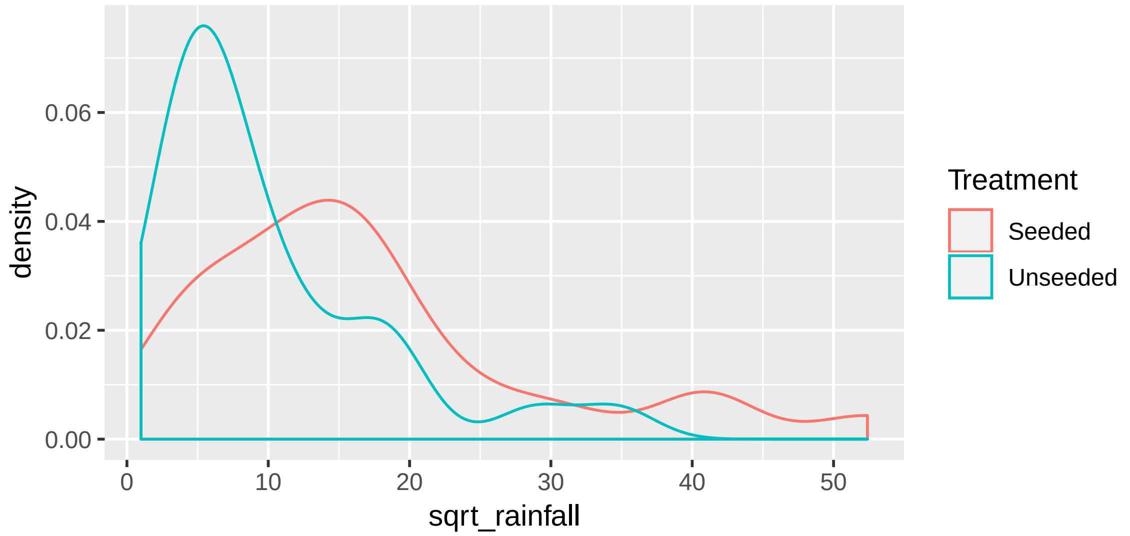
```
## # A tibble: 2 x 2
##   Treatment sd_rainfall
##   <chr>           <dbl>
## 1 Seeded           651.
## 2 Unseeded         278.
```

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)
  )
```

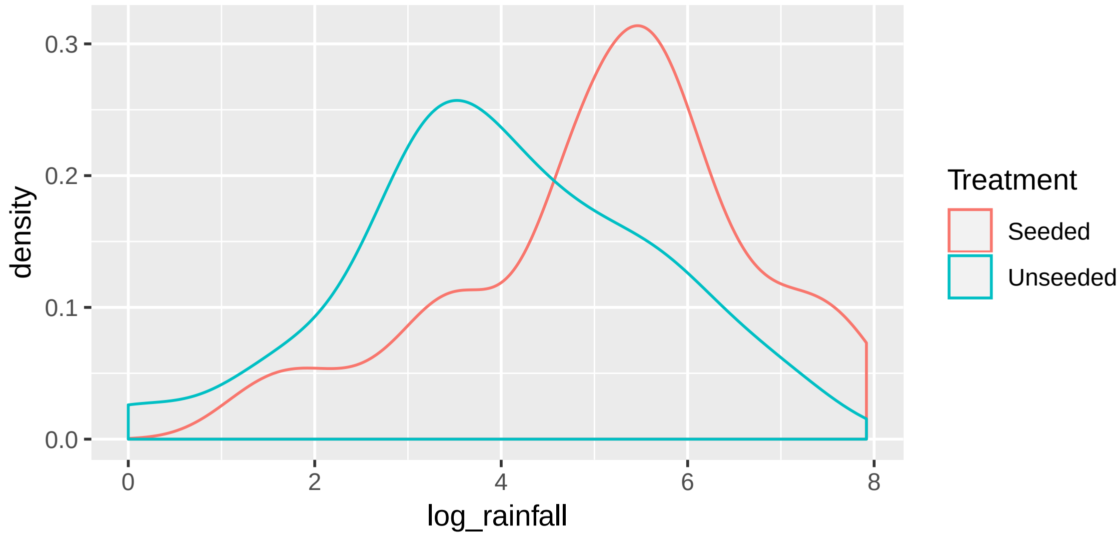
```
## # A tibble: 2 x 2
##   Treatment sd_rainfall
##   <chr>           <dbl>
## 1 Seeded          12.5
## 2 Unseeded         8.24
```

Still skewed right, 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

Separate group means, on log 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
```

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

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!):

```
exp(5.13)
```

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

```
exp(3.99)
```

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

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

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"
```

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

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
```