ST 411/511 Lab 3
Data Transformations

Instructions: Work through the following activity, either on your own, or along with the lab TA. You'll find last fall's video in the Media Gallery. Do the computational parts of Homework 3 shortly after finishing the lab activity.

Objectives for this Lab

- Do log and logit transformations.
- Locate data on a plot.
- Perform a t-test with some data excluded.

1. Download files Lab3.R and C_dalli.csv from Files>Lab Materials to your computer, noting the directory in which you save them. Start up RStudio and open Lab3.R. Load the Sleuth3 and ggplot2 R packages.

```
> library(Sleuth3)
> library(ggplot2)
```

2. In Chapter 3, we will consider the assumptions of t-tests and t-based confidence intervals. They are:

   - The populations are normally-distributed
   - When there are two populations, they have the same variance (i.e. the same standard deviation).
   - All observations are statistically independent.

3. We will start with the cloud-seeding case study of Chapter 3.

   (a) The `names()` command tells you what the columns of a data frame are called.

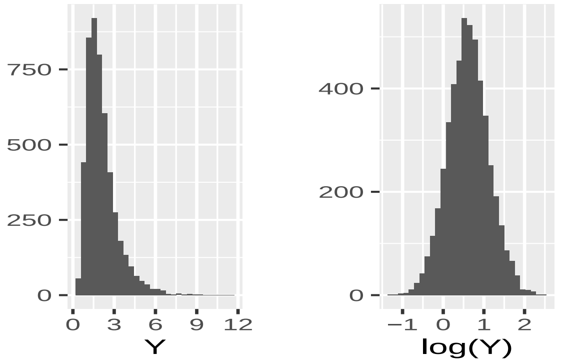
   ```
   > names(case0301)
   [1] "Rainfall"  "Treatment"
   ```

   (b) Produce a side-by-side boxplot of Rainfall.

   ```
   > qplot(Treatment, Rainfall, data=case0301, geom="boxplot")
   ```

   These distributions appear skewed with different spread, so we are not justified in making the assumptions that the populations are normal and the variances equal.

   (c) An easy and common remedy for right-skewed positive data is a log transformation. The log transformation is not always appropriate, but the pair of histograms below (from simulated data) illustrates the effect of a log transformation when it is.



In statistics, "log" always means "natural log," log base e, which mathematicians and engineers denote "ln." Sorry about that.

(d) It's important not to use a log transformation indiscriminately. You want the logged data to have symmetric distributions. Produce boxplots of the transformed data to check this.

```
> qplot(Treatment, log(Rainfall), data=case0301, geom="boxplot")
```

In this case, the log transformation seems to have solved our problems.

(e) Now that we are comfortable making the assumptions of a t-test, perform the test. If the research question is "does cloud-seeding **increase** rainfall?" then a one-sided t-test is appropriate (cf. item 6(e) of Lab 2).

```
> with(case0301, summary(Treatment)) # Check to see which group R puts first.
> t.test(log(Rainfall)~Treatment, data=case0301, var.equal=TRUE,
+         alternative="greater")
```

In `t.test()` set `alternative="greater"` because R orders the seeded group before the unseeded group. The alternative hypothesis is that the mean log acre-feet of rainfall is *greater* on the seeded days.

Your output should tell you that there is strong evidence that the population mean log acre-feet of rain is greater for the treated than the untreated days (one-sided $p = 0.007041$).

"Log acre-feet" are not natural units for these data, so as a courtesy to those reading your summary of statistical findings, you should report the results in the original "acre-feet" units. In lecture, we'll discuss this issue in more detail. Briefly, in reporting the results of the t-test, all that changes is the population parameter: instead of mean log acre-feet, the parameter of interest is "median acre-feet." Section 3.5.2 of the *Sleuth* discusses this, and we'll spend several minutes of lecture time as well. For now, we'll just note that the appropriate summary would be "there is strong evidence that the median acre-feet of rain is greater for the seeded than the unseeded days (one-sided $p = 0.007041$)."

(f) Reporting a confidence interval in the original units requires a little more work, and we will set this issue aside for this lab, and just construct the confidence interval in the transformed units, transform back to the original scale, and leave the report for lecture. Look back at your `t.test()` output. It should give a 95% confidence interval for the difference in mean log acre-feet of rainfall as $(0.3904045, \infty)$. R gives a one-sided confidence interval to go with our one-sided t-test. To get the two-sided confidence interval, do a two-sided test.

```
> t.test(log(Rainfall)~Treatment, data=case0301, var.equal=TRUE)
```

You should see a two-sided 95% confidence interval for $\mu_S - \mu_U$ of $(0.240865, 2.046697)$, where μ_S is the population mean log acre-feet on seeded days and μ_U is the population mean log acre-feet on unseeded days. The point estimate of the difference in means can be calculated as the difference in sample means (also shown on the output): $5.134187 - 3.990406 = 1.143781$.

(g) This point estimate and the endpoints of the confidence interval are in log acre-feet units. To return them to acre-feet units, exponentiate using e as base, since it is the base of the natural logarithm: a 95% confidence is $(e^{0.2408651}, e^{2.0466973})$. This is the `exp()` function in R.

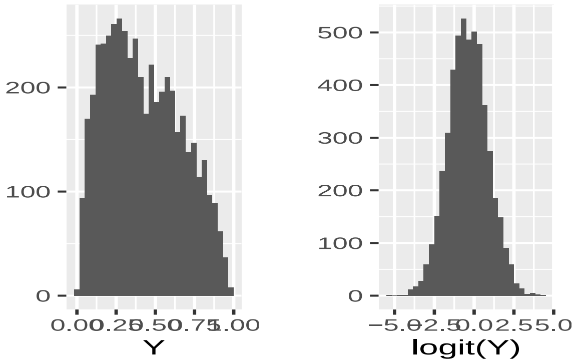
```
> exp(0.2408651)
> exp(2.0466973)
```

You should get a back-transformed confidence interval of $(1.272349, 7.742288)$. Back-transforming the point estimate should yield 3.138613. The point estimate is no longer in the center of the confidence interval because the exponential function has more of an effect on larger numbers.

4. The log transformation is very common, since many populations of interest are right-skewed. A useful transformation for data between 0 and 1 or percent data is the logit:

$$\text{logit}(Y) = \log\left(\frac{Y}{1-Y}\right) \tag{1}$$

The logit function is the inverse of the logistic function. The histograms below (from simulated data) illustrate the effect of a logit transformation on data bounded between 0 and 1, when such a transformation is appropriate.



The file C_dalli.csv contains percent cover data for *Conus dalli*, a species of sea snail, on four occasions. The researchers measured *percent cover* of the snail from photographs of experimental plots along the Oregon coast. For more information about this project, see marine.ucsc.edu/data-products/sea-star-wasting.

(a) Read the data into a data frame called `C.dalli`, and check the first few rows.

```
> C_dalli <- read.csv("C_dalli.csv")
```

(b) View the data in spreadsheet format.

```
> View(C_dalli)
```

(c) Have a look at boxplots by date.

```
> qplot(as.factor(time_point), pct_cover, data=C_dalli, geom="boxplot")
```

The `as.factor()`tells R to interpret `time_point` as categories, not numbers.

(d) Logit-transform the percent cover data, and save the data in a variable called `logit_pct`. Since `pct_cover` is given as values between 0 and 100, not 0 and 1, the transformation shown in equation (1) is

$$\text{logit}(Y) = \log\left(\frac{Y}{100 - Y}\right) \tag{2}$$

```
> logit_pct <- with(C_dalli, log((pct_cover)/(100-pct_cover)))
```

Check boxplots again.

```
> qplot(as.factor(time_point), logit_pct, data=C_dalli, geom="boxplot")
```

While the logit transformation seems to have resulted in more normal-looking data, it is not possible to provide a nice interpretation in the original units, as it was for the log transformation. Modern statistics has more appropriate models for percent data, but we will not see these in ST 411/511.

5. Now we consider the dioxin data of Case Study 3.1.2.

(a) Check the first few rows of the data frame, and produce boxplots.

```
> head(case0302)
> qplot(Veteran, Dioxin, data=case0302, geom="boxplot")
```

While these distributions are slightly skewed, they do appear to have about the same spread. According to the *Sleuth*, the only cause for concern is the presence of two extreme values. One strategy for dealing with these values is to compare results of analyses with and without these points.

(b) To identify the rows in `case0302` containing the two extremes, we can use R's `identify()` function. This function works with the `plot()` function in Base R, but not `qplot()`.

```
> plot(Dioxin~Veteran, data=case0302)
> with(case0302, identify(x=Veteran, y=Dioxin))
```

You should see a little stop sign at the top of the Console window, indicating that R is processing. In this case, it's waiting for you. Click on each of the points in the plot that you wish to "identify." When you're finished, hit the Escape key. The plot should now contain labels for the points you identified, and the Console should have output the row numbers.

(c) Perform a two-sample t-test to test $H_0 : \mu_V - \mu_N = 0$, where μ_V and μ_N represents the population mean dioxin concentrations for Vietnam veterans and non-Vietnam veterans, respectively.

```
> summary(case0302$Veteran) # Check R's ordering of the groups.
> t.test(Dioxin~Veteran, data=case0302, var.equal=TRUE, alternative="less")
```

(d) Repeat the test while omitting the most extreme observation #646.

```
> t.test(Dioxin~Veteran, data=case0302, var.equal=TRUE, alternative="less",
+         subset=-646)
```

(e) Repeat the test while omitting the two most extreme observations #646 and #645.

```
> t.test(Dioxin~Veteran, data=case0302, var.equal=TRUE, alternative="less",
+         subset=-c(646,645))
```

(f) Compare results of these three tests with each other and with Display 3.7 on page 70 of the *Sleuth*.