Statistics 411/511
Lab 3

Lab Instructions: If you want to work along with the TA in lab, please sit near the front. If you prefer to go to lab but work at your own pace, please sit near the back, and wait for the appropriate time to ask any questions.

Lab 3: Data Transformations

Objectives for this Lab

- Do log and logit transformations.
- Read in a data set from a .csv file.
- Locate data on a plot.
- Perform a t-test with some data excluded.

1. As usual, start up RStudio. Load the Sleuth3 and ggplot2 R packages. If you are working in Bexell, you'll have to install the packages again as described in item 5(a) of Lab 1.

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

2. Download Lab3.r to your Z: drive, then open Lab3.r in RStudio (File>Open file...).

3. Set your working directory to match the directory containing Lab3.r. This will be important below when we read in a data set from a file, rather than using data from the Sleuth3 package.

   From the Session menu, select Set Working Director>To Source File Location. RStudio will enter a `setwd()` command in the Console for you.

4. Let's start with the cloud-seeding case study.

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

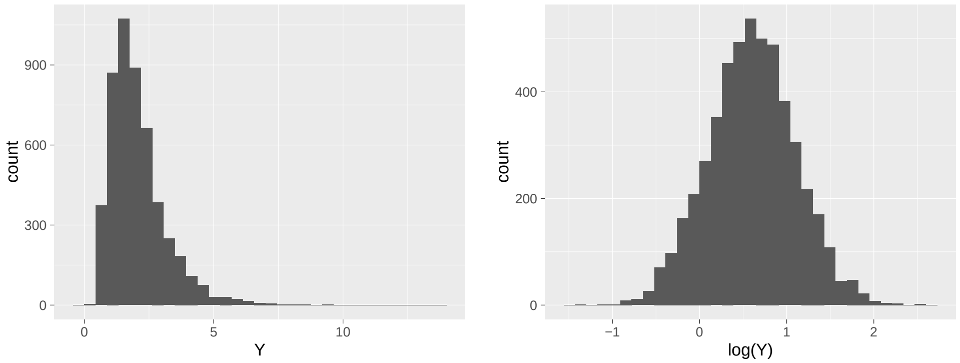
   ```
   > names(case0301)
   ```

   (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 illustrates the effect of a log transformation when it is.



(d) Take the (natural) log of Rainfall and call the log-transformed data `log.rainfall`.

```
> log.rainfall<-log(case0301$Rainfall)
```

(e) 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.

(f) 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).

```
> summary(Treatment) # Check to see which group R puts first.
> t.test(log.rainfall~Treatment,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 (true 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, report the results in the original "acre-feet" units. 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" (see Section 3.5.2 of the *Sleuth*). Thus, an appropriate sentence would be "there is strong evidence that the median acre-feet of rain is greater for the seeded than the unseeded days (one-sided $p = 0.007041$)."

(g) Reporting a confidence interval in the original units requires a little more work. But first, 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 $(-\infty, -0.3904045)$. 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,var.equal=TRUE)
```

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

(h) 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)
```

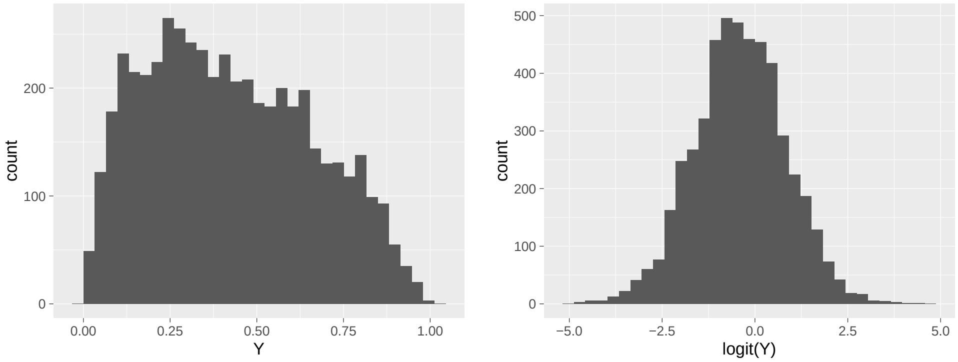
You should get a back-transformed confidence interval of $(1.272349, 7.742288)$. Back-transforming the point estimate should yield 3.138613.

Because a log scale is multiplicative, the interpretation of the point estimate and confidence interval on the back-transformed scale is about the *ratio* of the medians. Details are in Section 3.5.2 of the *Sleuth*. The "interpretation after log transformation" on page 72 of the *Sleuth* applied to the back-transformed point estimate and confidence interval gives the statement in the summary of statistical findings on page 57.

5. You may need to make other data transformations. A useful transformation of 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 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 http://www.bco-dmo.org/project/517379 and http://www.eeb.ucsc.edu/pacificrockyintertidal/data-products/sea-star-wasting/.

(a) Download this file from the Files>Lab Materials on Canvas and save it on your Z: drive. This should be your working directory (see item 3 above).

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

```
> C.dalli <- read.csv("C_dalli.csv")
> head(C.dalli)
```

(c) View 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. Since pct_cover is given as values between 0 and 100, not 0 and 1, the transformation (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 improved the normality of the data, it is not possible to provide a nice interpretation in the original units, as it was for the log transformation.

6. 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 extreme, we can use R's `identify()` function. This function works with the `plot()` function but not `qplot()`.

```
> plot(case0302$Dioxin)
```

The result is a scatterplot of the dioxin measurements vs. "Index," the row in the data frame. If there were only 10 or 15 rows, we could pick off the row numbers of the extreme points from this plot. Here, there are almost 750 rows, so we use `identify()` to find the rows numbers of the two points with extremely high dioxin readings.

```
> identify(case0302$Dioxin)
```

You should see a litte 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*.