Introduction to Biostatistics (171:161)
Breheny

Lab #13

In the previous lab we began analyzing Infant Diarrhea study, in lab #13 we will further anaylze that data set using what we now know about outliers, transforming data and non-parametric procedures.

1 Infant diarrhea study

We will begin by revisting the distribution of the Infant Diarrhea data. Pay particular attention to the strong right-skew and the outliers.

SAS:

```
PROC SGPLOT DATA=diarrhea;
  VBOX Stool / CATEGORY=Group;
RUN;
```

R:

```
boxplot(Stool~Group,col="gray")
```

2 Outliers

We know that extreme observations have an unfair impact on the mean. Since the two-sample t-test is based on comparing the means within each study group, anything that impacts the mean will impact the results of the test. We also know that outliers influence the standard error which will also factor in towards the test results. Lets try using 750 ml/kg as an upper limit to remove the outliers in each group. Compare the results we get from this to analysis from lab #12 when the outliers were included.

In SAS, we can work with subsets of our data using the `WHERE` option; in `R`, `diarrhea[condition,]` tells `R` to use only those rows the data set `diarrhea` where `condition` is true:

SAS:

```
PROC TTEST DATA=diarrhea;
  WHERE Stool < 750;
  CLASS Group;
  VAR Stool;
RUN;
```

R:

```
t.test(Stool~Group,
       diarrhea[Stool < 750,],
       var.equal=TRUE)
```

Note that after throwing out these observations the p-value is a lot lower. It is important to recognize two points here:

- The *t*-test is quite heavily affected by those severe cases, for which the severity of the illness was much higher than the average. This might not be desirable, to have your analysis so highly affected by such a small percentage of your data. Perhaps we should do something about it.
- Okay, we should do something about it, but arbitrarily throwing away some of the data is not the ideal solution! In particular, if you're trying to say something about whether bismuth salicylate can help sick infants, how can you justify excluding the sickest of these children?

3 Log transformations

One approach that would be better than throwing away data is to take a log transformation of the original response. As we discussed in lecture log transformations work well when the data is right skewed.

SAS:

```
DATA diarrhea;
  SET diarrhea;
  logStool=log(Stool);
RUN;
```

R:

```
logStool <- log(Stool)
```

SAS:

```
PROC SGPANEL DATA=diarrhea;
  PANELBY Group;
  HISTOGRAM logStool;
RUN;
```

R:

```
require(lattice)
histogram(~logStool|Group)
```

From the plot of `logStool`, you can see that there's still a little bit of skewness to the data, but far less than there was originally. We can now perform a *t*-test on `logStool` and acheive a more powerful result.

SAS:

```
PROC TTEST DATA=diarrhea;
  CLASS Group;
  VAR logStool;
RUN;
```

R:

```
t.test(logStool~Group,
       var.equal=FALSE)
t.test(logStool~Group)
```

Note that our *p*-value is four times lower than before. We've lessened the *impact* of those extreme cases without actually removing them from our data set.

Also note the confidence limits provided are on the log scale. In order to obtain a more interpretable interval we need to exponentiate them.

4 Wilcoxon rank-sum test

Another approach that is also better than throwing away outliers is to use a rank-based method like the Wilcoxon rank-sum test (also known as the Mann-Whitney test). The rank-sum test is analagous to a two-sample t-test.

The procedure for doing this is SAS is the rather awkwardly named `PROC NPAR1WAY`; the "npar" part refers to the idea that this is a nonparametric approach ... unfortunately there is not a short explanation for the "1way" part of the name. In `R`, this test can be carried out via `wilcox.test`. These procedures have exactly the same syntax as the t-test procedures:

SAS:

```
PROC NPAR1WAY DATA=diarrhea;
  CLASS Group;
  VAR Stool;
RUN;
```

`R`:

```
wilcox.test(Stool~Group)
```

SAS outdoes itself here and gives you about 10 times more information than you really want. Hunt through the pages of output until you come to the part about the "Wilcoxon Two-Sample Test". When people refer to the "Wilcoxon rank-sum test", they generally mean the one that uses the normal approximation. So, in this case, we would get a p-value of .0056 (which is almost identical to the p-value we got from analyzing the log-transformed data).

Note about the different "Wilcoxon" tests: SAS offers several different approximations of the exact answer, any of which are fine to use in this class. The exact answer can be obtained using an `EXACT WILCOXON` statement. The same can be done in `R` using `exact` is TRUE.

5 Wilcoxon signed-rank test

If the data you are analyzing is matched/paired the signed-rank test is a non-parametric procedure that takes in to account the relatedness between the two groups. The signed-rank test is analagous to a paired t-test.

To do the Wilcoxon signed-rank test in SAS you need to create a variable for the difference between the two groups. `PROC UNIVARIATE` will do the test and provide the results. In `R`, you can do the test using the `wilcox.test` function with the argument `Paired = TRUE`

As an example we will use the oat bran and corn flakes cholesterol data that we have previously looked at in several homework assignments.

SAS:

```
DATA oatbran;
SET oatbran;
diff = cornflakes - oatbran;
RUN;
PROC UNIVARIATE DATA = oatbran;
  VAR diff;
RUN;
```

R:

```
wilcox.test(CornFlakes, OatBran, paired = TRUE)
```

Recall that the paired t-test resulted in a p-value of 0.005, compare this to results from the above approach.

Note: `R` does an exact test here when n<50 and approximate test otherwise. SAS does an exact test when n<20.

The bottom line of this lab: when you've got continuous data, don't just blindly apply a t-test. Look at the data, and if it's skewed or contains large outliers, consider a transformation or a rank-based method instead.