TRANSFORMING DATA TO BETTER FIT ASSUMPTIONS

Were Times for some copter variables (e.g., Group?) not normal? If so, this can get
awkward. One option is to transform data (e.g., logX, or sqrtX), but you cannot
transform one set (e.g. $\log_{10}$(Time) for Design 31) while leaving another untransformed.
Otherwise, you will be comparing very different values (e.g., 10 vs. $\log_{10}(10) = 1$) and find a
significant difference for designs merely because you converted some.

IMPORTANT: If you transform a variable, you have to do so for all groups to be compared.
ALSO: Be very careful with back-transforming data to report means, etc. The log of a mean is
not the same as the mean of a log.

But transforms can make distributions normal and variances among sets homogeneous.
Or nearly so. And so transformations are common in parametric statistics – you may already
have seen papers reporting log or square-root versions of data. A second option is to use more
advanced methods (we get to those in a few weeks). Meanwhile, try this to get used to the idea:

Import the copter data again, from
https://sciences.ucf.edu/biology/d4lab/wp-content/uploads/sites/125/2019/09/helicopter-data.txt
Here we assume you called it "data"

select a Group – you choose which one. Maybe a skewed one. and plot a histogram of Time
Now compute two transformations of that same data (e.g., dataW below), like this (same as in excel, etc.)

```
logTime <- log10(dataW$Time) # notice that we specify log base-10. "log" is the natural log.
srTime <- sqrt(dataW$Time Time) # square root transformation of the data
```

now make boxplots of those data and compare to the original: More normal? Worse?

Now look at the skewness plots in the Introduction here:
http://en.wikipedia.org/wiki/Skewness

Do some of our copter data look like these? If so, you can fix it with transformations.
This is similar to using different measures (e.g., meters vs. feet) for the same object.
If you think about it that way, transformations are no big deal …

Here are some guidelines, where we name the new variable newX for any original variable X:

# If your data have	Use this transformation
# Moderately positive skewness	Square-Root - e.g., newX = sqrt(X)
# Substantially positive skewness	Logarithmic - e.g., newX = log10(X)
# Substantially positive skewness	Logarithmic - e.g., newX = log10(X + C)
# (with zero values)	where C is a constant, often 1
# Moderately negative skewness	Square-Root - e.g., newX = sqrt(K – X)
#	where K is a constant, often max(X)+1
# Substantially negative skewness	Logarithmic - newX = log10(K – X)
#	where K is a constant, often max(X)+1

Let's calculate some transforms. Compute a log10 and a square-root transformation of **all** Time
data.

```
lg <- log10(Time)
sr <- sqrt(Time)
```

Both of these variables appear in the Environment window (upper right), but are not yet
combined with our data. That means our subset command won't work on them yet.

Combine these transformations with your data file by using `cbind` to "bind columns":

```
data <- cbind(data,lg,sr)
```

Now repeat histograms, and then run normality and homoscedasticity tests (like you did in the
last lab) to evaluate plain Time vs. log-transformed Time vs. square-root-transformed Time.

Did a log- or square-root-transform help make data fit better to our assumptions?
If so, then you can expect to use that transformation in subsequent analyses.

You might already imagine this iterative process for each variable in a large study is a big
nuisance and can be frustrating if it works for some groups but not others. That's why more
sophisticated analyses that permit other distributions (e.g., negative binomial) and relax
assumptions of normality and homoscedasticity are verrrrrry convenient, much more robust,
and often better detect effects you study. But those will come later ...