Statistical Literacy for Biologists: Week 4, Part 1

Frankline M. Onchiri

20 October 2017

Treat statistics as a science, not a recipe. ~ Andrew Vickers

Transformations: What, When, Why, and How

During the lecture on distributions, you learned about selecting an appropriate distribution for your response variable, which helps identify the best model or set of potential models to address your research question. It is possible to transform either your response variable or your predictor variable(s) when building a model in order to meet distributional assumptions underlying the models or to conform with the form of relationship between outcome and predictor. **Transformation** is the re-expression of data on a new scale using a mathematical function for each data point. Before we look at specific transformations let's look at the general model construction process.

Model Building and Use

Constructing a model is often iterative. The basic approach can be summarized in the following steps.

1) Model Building
   a) Establish the main scientific research question and **refine scientific hypotheses into statistical hypotheses (before looking at the data)**.
   b) Formulation *based on scientific question, study design, carefully selected variables and distributional properties of outcome variable.*
   c) Estimation *of parameters; measures of associations of outcome with indendent variables.*
   d) Evaluation: *Adequacy of model; how well model fits/summarizes data. Extent to which data meets technical requirements of statistical procedure.*
2) Model Use: *Test hypotheses about the parameters to answer the research question of interest.*

Ultimately we want a model that meets **underlying assumptions** and answers our research question(s). Examples of such assumptions include: distribution (e.g. Normality, binomial, poisson), independence of observations, same variance for every value of the predictors, and shape of the relationship between outcome and main predictor of interest (Linear or non-linear). **Violations of these assumptions may result in misleading results; biased estimates of regression coefficients and undermine the validity of CIs and P-values**. Outside of prediction, we also would prefer a model that is **parsimonious-** *one that addresses the research question with fewer independent variables of scientific interest.*

We don't move onto using a model until we've passed through the modeling building approach and have satisfied any necessary assumptions. We might have to **transform** the **outcome variable** in order to make the assumptions approximately hold. Besides, depending on the form of the relationship, we might also have to transform the **predictor variable**, or **both outcome and predictor**. Today we will focus on transformations of the response variable.

Reasons to consider transforming variables

"*Transformations are needed because there is no guarantee that the world works on the scales it happens to be measured on*" ~ Boston College FMRC web

1) Meet distributional assumptions, e.g. normality (bell-shaped) assumption.
2) Stabilize spread/variances: equalize variances across comparison groups.
3) Simplify the the form of relationships between variables; e.g. to linearize - making a non-linear relationship linear.
4) Improve predictions and better diagnostics of fit and residuals (as in regression analysis).

Also, we can improve interpretations when we find alternate ways to understand the data and discover patterns or relationships that may not have been revealed on the original scales.

How do you know when to transform or what transformation?

1. Graphical visualizations (e.g. scatterplots with superimposed curves), techniques of exploratory data analysis
2. Theoretical basis - what makes scientific or biological sense
3. Eperimentation with different transformations and use of R^2 or AIC to assess models

Manuscript: *Improving Student Performance in Organic Chemistry: Help Seeking Behaviors and Prior Chemistry Aptitude. Horowitz G. et al. (2013); Journal of the Scholarship of Teaching and Learning, Vol(13)*

Reviewer's comment: *How did you choose the particular transformations you used?*

Author's response: *Transformation is the replacement of a variable by a function of that variable (e.g., replacing variable x by the square root or logarithm of x) that changes the shape of a distribution* ***so that data more closely meet the assumptions of a chosen statistical inference procedure****. We chose a square root transformation (commonly used for positive data in the behavioral and social sciences)* **because it worked well with our data**.

General Guidelines for Transformations

Transformations can be undertaken when necessary to help meet modeling assumptions, e.g., to address non-linearity and/or to stabilize variance. ***Transformations are most appropriate when they match a scientific view of how a variable behaves***. It is generally better to sacrifice some goodness of fit and keep a clear interpretation. Unless data values are known errors you should not remove values to meet distributional assumptions with or without transformations.

Commonly used transformations for outcome variables

Continuous outcome data

Logarithmic (log) transformation is the most common transformation. When we say log transformation without any additional information it is assumed to be the natural logarithm.

$$y_{new} = ln(y)$$

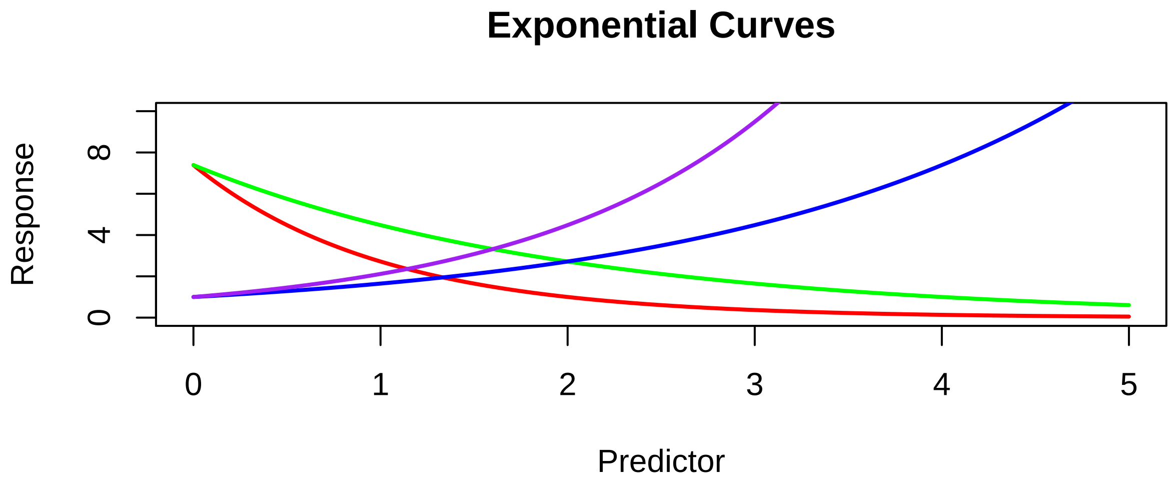
the transformed outcome data is then analyzed using linear regression.

Examples: In biomedicine, logarithmic transformations are commonly applied to measurements of biological processes e.g. antibody concentration and mRNA concentration (gene expression), because these concentrations differ by orders of magnitude across individuals (and sometimes within individuals over time). Similarly owing to the *exponential growth* associated with viral replication, viral load is generally analyzed after logarithmic transformation.

In some cases, scientific mechanisms dictate that ratios are better summary measures for effects due to predictors. e.g. when exposures/interventions affect the rate that disease occurrs over time.

Log transformations are also used to transform data in the form of *exponential growth or decline*. Below are some examples of exponential relationships.

```
set.seed(seed=792)
# Generate a regular sequence of numbers
xseq1<-seq(from=0, to=5, by=.01)
plot(x=xseq1, y=exp(2-xseq1), col="red", xlab="Predictor", ylab="Response", ylim=c(0,10), type="l", lwd=
lines(x=xseq1, y=exp(2-0.5*xseq1), lwd=2, col="green")
lines(x=xseq1, y=exp(0.5*xseq1), lwd=2, col="blue")
lines(x=xseq1, y=exp(0.75*xseq1), lwd=2, col="purple")
```



As you saw previously the lecture on distributions, *right-skewed* data may be transformed closer to normality by log transforming the values. If negative values are present, an alternative is the *reciprocal transformation*

$$y_{new} = 1/y$$

or

$$y_{new} = -1/y$$

.

Dealing with 0s in Transformations

The logarithmic transformation cannot be used directly for data that includes zeroes. Sometimes the transformation

$$ln(x+1)$$

is used so that transformed these values will still have value zero.

Recommendations are similar when dealing with zeroes and the reciprocal transformation.

Discrete non-negative quantitative outcome data

If the response variable is a **count**, the variances of the error terms are not constant but rather depend on the value of the predictor.

A commonly used, but now outdated, recommendation is to transform the response variable using the *square root transformation*:

$$y_{new} = \sqrt{y}$$

and then use the *linear regression* framework.

Now it is more common to use count regression methods, based on either the Poisson or negative binomial distributions as we discussed last week. When the regression coefficients of Poisson or negative binomial regression models are exponentiated they yield rate ratios (**We'll discuss more later**).

Binary outcome data If the response is a binomial proportion, the **variances of the error terms also are not constant** but depend on the value of the predictor. A commonly used, but now outdated, recommendation is to transform the response using the *arcsine transformation*

$$y_{new} = arcsin(y)$$

and then use the *linear regression* framework. Arcsine is the inverse of the sine function you'll recall from trigonometry.

Now it is more common to use a regression method based on the binomial distribution called **logistic regression**. When the regression coefficients of logistic regression models are exponentiated they yield odds ratios.

Note: The reciprocal *reverses* order among values of the same sign: *largest becomes smallest*, etc. The negative reciprocal preserves order among values of the same sign

Displaying vs. Modeling Transformed Data

As discussed at the end of the last class, when plotting to assess model fit when we have predictors, we work with the estimates of the error terms in the model, the residuals.

How do I interpret transformed data?

A major limitation of some of the transformations is that they make the data less interpretable, you almost always want to get back to the original measurement scale to answer your research question. **Another complication is that of explaining to the non-statistician why you are modeling the square root of their favorite variable rather than the variable in its unadulterated form**. It is easier to "live with" invalid models with strange looking residuals that nobody cares about than it is to explain the complexities of the analysis.

Statistics humour: ***A physician makes an analysis of a COMPLEX ILLNESS whereas a statistician makes you ILL with a COMPLEX ANALYSIS***

Alternatives to Transformations

The ordinary linear regression model for continuous, normal data is usually referred to as the *General Linear Model*. In any regression model, a function of some summary measure (commonly represented using greek letters μ or θ) of the outcome variable is modelled as a linear function of the predictors. In linear regression, it is the *average/mean* of the *continuous* outcome. Data from binary outcomes are summarized using *proportions(p)* (so here θ=p), which are restricted within the range of (0, 1). Data in the form of *counts* have to be positive always, and are often summarized using *rates* (θ=rate). To handle these data properly, we must take account of the *bounded nature* of the response. Using the ordinary linear regression to model fractional/proportion, or other bounded summary measures can results in predictions outside the closed interval [0-1]. Besides, we have already seen that the mean and variance of binary or count data are connected. Therefore they don't have constant variance. In addition, some transformations such as squareroot, reciprocal or arcsine may make interpretations and explanations of the model results difficult.

Because of these limitations, bounded data are analyzed using **General*ized* Linear Models (GLMs)** approach. The GLMs are alternatives to transformations. In GLMs, a mathematical function (*link function*, because it "links the summary measure to a linear combination of predictors") is used to transform a bounded summary measure of **non-normal** data into **unbounded "continuous"** data that are linearly related to predictors. This way, we can still use the familiar **one-unit change** language to describe the effects of predictors the average outcome. *Thus, the GLM assesses the linear predictor on an UNBOUNDED TRANSFORMED SCALE.* The use of an appropriate link function ensures that the predicted values of the outcome variable, when converted back into the scale of original outcome, will stay within the boundaries of the observed data (e.g. predicted probabilities for binary outcomes will be restricted to 0/1).

The general form of a generalized linear model is:

$$g(\theta) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n$$

For outcomes that are already normal, the general linear model is a special case with an *identify* link function ($\theta * 1$).

GLM transformation of count or binary have natural *rate ratios* and *odds ratios summaries*, respectively.

Example: For **Binary outcome data** the link function is the natural logarithm of the odds (often referred to as the *Logit Function*) so that the new unbounded transformed outcome variable is

$$y_{new} = ln(p/1-p) = ln(\theta/1-\theta),$$

which is the natural logarithm of th odds. While the range of the original θ is 0/1, the range of θ_{new}=$logit(p)$ is unbounded and goes from -∞ to +∞. The transformed outcome [*logit(p)*] is then linearly associated to the predictors. Predictor effects are linear and additive like in GLM, and the coefficients are interpreted as change in logit(y) per one-unit change in predictor A Logit link is a nonlinear transformation of probability. The model will be linear with respect to the predicted logit, which translates into a nonlinear prediction with respect to probability of the outcome which will be bounded between 0 and 1 as needed.

The GLM for binary data then becomes $E(g(y = 1)) = logit(p) = \alpha_0 + \alpha_1 X_1$. Exp($\alpha_1$) has the interpretation of the odds ratio of the outcome association with **one-unit change** in the predictor variable.

$$g(\theta) = logit(\theta) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n$$

For continuous variables θ is the mean, and θ is the proportion, and rate for binary and count data respectively

The table below summarizes the different types of GLMs based on the type of outcome variable:

Distribution	Link Function	GLM in R	Algebraic form
Normal	Identity	glm(y~x, family=gaussian)	$g(\theta) = \beta_0 + \beta_1 X_1$

Distribution	Link Function	GLM in R	Algebraic form
Lognormal	Identity	glm(log(y)~x, family=gaussian)	$log(\theta) = \gamma_0 + \gamma_1 X_1$
Binomial	Logit	glm(y~x, family=binomial)	$logit(\theta) = \alpha_0 + \alpha_1 X_1$
Poisson	Log	glm(y~x, family=poisson)	$log(\theta) = \lambda_0 + \lambda_1 X_1 + log(t)$
Negative binomial	Log	glm.nb(y~x)	$log(\theta) = \theta_0 + \theta_1 X_1 + log(t)$

Note: Remember for the Poisson distribution, the outcome is a count (whole numbers greater than or equal to 0), and is characterized by a single parameter, which is the mean rate of occurrence for outcome of interest. The mean and variance are the equal. When the variance of count data is larger than mean, there is **overdispersion**. Performing Poisson regression on count data that exhibits this behavior results in a model that doesn't fit well. In this case an alternative distribution called the Negative Binomial Distribution is used. Unlike the Poisson distribution, the variance and the mean are not equivalent. Thus it might serve as an ideal distribution for modeling counts with variability different from the mean.

If you want to fit a model using *glm* with no predictors the syntax is ~1 in place of the predictors.

We are going to look at some regression models today that build on the distribution work from last week. To fit these models we will need the package MASS.

```
library(MASS)
```

Example 1

Let's revisit the viral count data from last week.

```
# Simulate some data to represent the underlying viral load
viralLoad<-c(rep(0,265), runif(78,200,200000), runif(110,200001,1000000), runif(40,1000001,5000000), ru
```

Let's try fitting a logistic regression model for any viral load.

```
viralLoadPos<-ifelse(viralLoad > 0, 1, 0)
table(viralLoadPos)
```

```
## viralLoadPos
##   0   1
## 265 235
```

```r
vLPfit<-glm(viralLoadPos~1,family=binomial)
summary(vLPfit)
```

```
##
## Call:
## glm(formula = viralLoadPos ~ 1, family = binomial)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.127  -1.127  -1.127   1.229   1.229
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.1201     0.0896  -1.341     0.18
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 691.35  on 499  degrees of freedom
## Residual deviance: 691.35  on 499  degrees of freedom
## AIC: 693.35
##
## Number of Fisher Scoring iterations: 3
```

```r
# residuals(vLPfit)
# fitted.values(vLPfit)
```

Exercise 1

Fit a logistic regression model for smoking status in the systolic blood pressure data set.

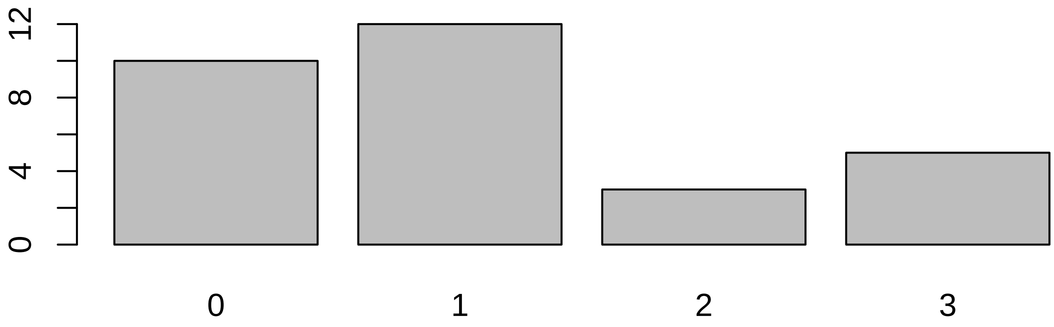
```r
# Load sbp data from earlier weeks
sbp = read.csv("https://ssc.ca/sites/ssc/files/archive/documents/case_studies/2003/documents/datafile.da
sbp = sbp[ , 1:18]
sbp[ , c(2:5, 9:12, 14:18)] = data.frame(apply(sbp[ , c(2:5, 9:12, 14:18)], 2, as.factor))
str(sbp)
```

```
## 'data.frame':    500 obs. of  18 variables:
##  $ sbp     : int  133 115 140 132 133 138 133 67 138 130 ...
##  $ gender  : Factor w/ 2 levels "F","M": 1 2 2 2 2 1 1 1 2 2 ...
##  $ married : Factor w/ 2 levels "N","Y": 1 1 1 2 1 1 2 2 2 2 ...
##  $ smoke   : Factor w/ 2 levels "N","Y": 1 2 2 1 1 1 1 1 1 2 ...
##  $ exercise: Factor w/ 3 levels "1","2","3": 3 1 1 2 2 3 1 3 1 3 ...
##  $ age     : int  60 55 18 19 58 55 22 52 46 38 ...
##  $ weight  : int  159 107 130 230 201 166 188 123 106 166 ...
##  $ height  : int  56 65 59 57 74 67 66 67 73 72 ...
##  $ overwt  : Factor w/ 3 levels "1","2","3": 3 1 2 3 2 2 3 1 1 1 ...
##  $ race    : Factor w/ 4 levels "1","2","3","4": 1 1 1 2 1 1 1 1 1 1 ...
##  $ alcohol : Factor w/ 3 levels "1","2","3": 2 2 1 3 3 1 3 2 3 1 ...
##  $ trt     : Factor w/ 2 levels "0","1": 1 1 1 2 1 2 2 1 2 2 ...
##  $ bmi     : int  35 17 26 49 25 25 30 19 13 22 ...
##  $ stress  : Factor w/ 3 levels "1","2","3": 2 2 3 3 2 2 3 2 2 2 ...
##  $ salt    : Factor w/ 3 levels "1","2","3": 2 2 2 3 2 1 1 3 2 2 ...
##  $ chldbear: Factor w/ 3 levels "1","2","3": 2 1 1 1 1 3 3 2 1 1 ...
##  $ income  : Factor w/ 3 levels "1","2","3": 2 3 1 1 2 2 1 3 2 1 ...
##  $ educatn : Factor w/ 3 levels "1","2","3": 2 2 3 2 3 3 1 2 1 1 ...
```

Example 2

Let's revisit the tumor count data from earlier weeks.

```
# Create some fake count data with mean = 1
set.seed(54)
tumor_count = data.frame(x = rpois(30, 1))
barplot(table(tumor_count$x))
```



Let's fit a Poisson regression model.

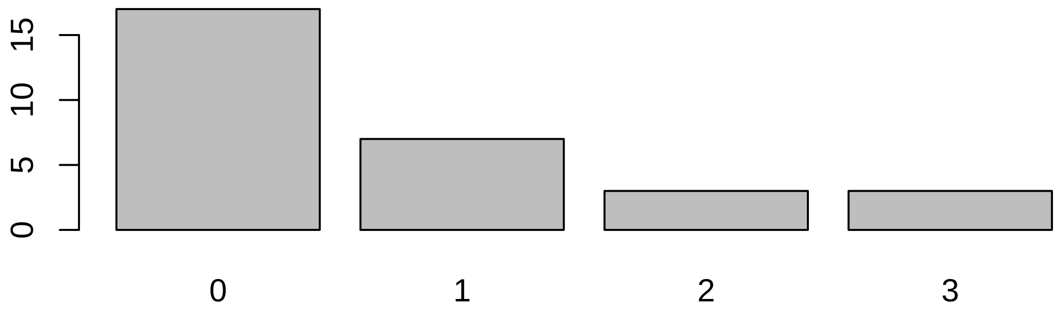
```
summary(glm(tumor_count$x ~ 1, family=poisson))
```

```
##
## Call:
## glm(formula = tumor_count$x ~ 1, family = poisson)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.48324  -1.48324  -0.09685   0.55253   1.48990
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.09531    0.17408   0.548    0.584
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 34.986  on 29  degrees of freedom
## Residual deviance: 34.986  on 29  degrees of freedom
## AIC: 83.786
##
## Number of Fisher Scoring iterations: 5
```

Exercise 2

Fit Poisson and negative binomial regression models for the example data below.

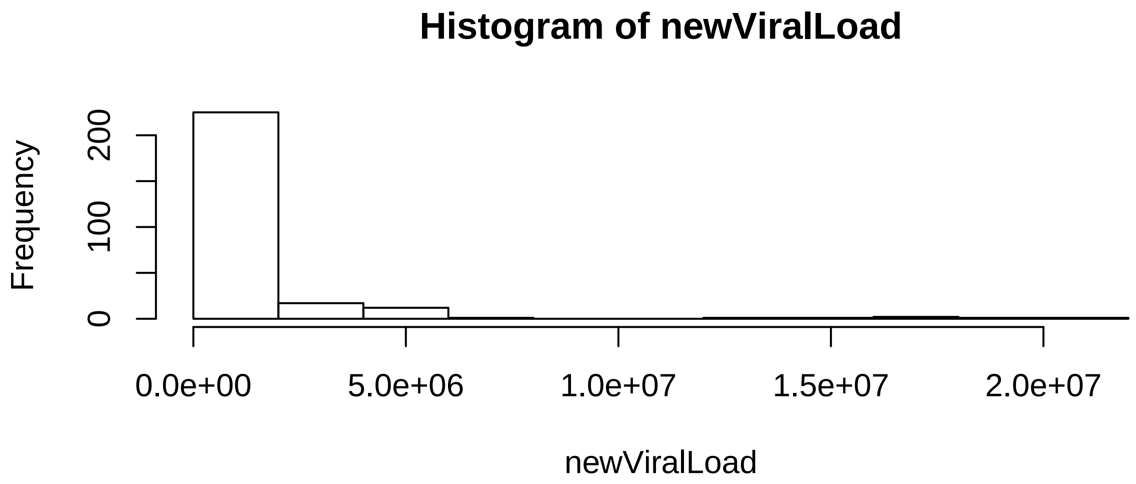
```
set.seed(45)
new_tumor_count = data.frame(x = rnbinom(30, prob=0.6, size=1))
barplot(table(new_tumor_count$x))
```



Homework

1. Fit a negative binomial regression model to the viral count data found below.

```r
newViralLoad<-c(rep(0,26), runif(78,200,200000), runif(110,200001,1000000), runif(40,1000001,5000000), 
hist(newViralLoad)
```



2. Fit a logistic regression model to the systolic blood pressure data, using as the outcome high systolic blood pressure (yes/no), where the cutoff for high is >140.

End of file