# A Study of Generalized and Vector Generalized Linear Models

Sam Lorenz

A thesis presented in partial fulfillment of a Bachelor of Arts with College Honors in Data Science at Knox College, Galesburg, Illinois.

May 2023

**College Honors Committee**:

Ole J. Forsberg, Committee Chair
Associate Professor of Mathematics
Chair of Statistics Program

Mary V. Armon
Associate Professor of Mathematics

John Haslem
Director, Center for Teaching and Learning

Benjamin G. Solomon, Outside Examiner
Associate Professor of Educational & Counseling Psychology
University at Albany

## Abstract

In many real-life applications, modeling using the classical linear modeling paradigm (CLM) will not suffice due to violations of its requirements, namely that the response variable has a conditionally Normal distribution. When this requirement is not met, one usually opts for using a generalized linear model (GLM). GLMs require that the response variable has a conditional distribution in the exponential family of distributions. Should this requirement also be violated, we can perform an extension of the GLM— a vector generalized linear model (VGLM). In this paper, we will discuss some of the key similarities and differences between these modeling paradigms by comparing the models in R with a data set and studying the mathematical foundation of CLM, GLM, and VGLM.

Statistics has become a booming career choice in recent years, especially due to the progression of technology and modeling. In statistics courses at Knox College, I have had experience modeling using the classical linear model (CLM), the generalized linear model (GLM), and the vector-generalized linear model (VGLM). I have had practice using each of these models, but the reason to use them as well as the specialties of each model is something I have yet to fully understand.

The purpose of this research is to better understand the three different types of modeling. The CLM and the GLM are older than the more recent VGLM. The CLM has a list of requirements for the data before the model can be estimated. When the data actually meet the requirements (which is quite rare in real-life applications), then it is a simple model to use and understand. In the case that the data follow a distribution in the exponential family, it is a clear choice to use the GLM instead. The GLM has fewer requirements and the data does not need to be forced to fit the requirements of the CLM.

There are times when the data do not follow a distribution in the exponential family and thus violate the requirements for the CLM; furthermore, a researcher may need to model more than one parameter. In this case, the VGLM is an incredibly versatile model that allows a statistician to model data outside of the exponential family and specify variables for more than one linear predictor.

Aside from researching the mathematics and definitions of these models, I saw another important aspect was comparing the models using R to understand aspects of their application. Two examples to note the two important characteristics of the VGLM will be conducted in R. The first example utilizes an election in a fantasy kingdom when we use a distribution outside the exponential family. The second will be a fictitious battle that requires modeling

more than one parameter. To wrap up the differences and similarities, I completed a real-life application with data from Professor Solomon to understand how students improve their reading skills over a time period. Finally, there will be final comments and reflections on the different types of modeling.

# 1 All Things Statistical Modeling

In this section, the three primary types of regression modeling will be discussed: Classical Linear Models, Generalized Linear Models, and Vector Generalized Linear Models. The discussion considers how they differ from one another, the requirements of each type of modeling, how they are used, and how the parameters are estimated.

## 1.1 The Classical Linear Model

I enjoy reading fantasy novels. Specifically, I like young authors' fantasy the best, because such works feature easy plots to get lost in without much romance, violence, and other things usually found in adult fantasy books. The best fantasy books to read include plots where a kingdom is built, an army is trained, a lot of magic is used, and the ending is happy.

To illustrate key components of the CLM, suppose we have a Queen in a fantasy novel concerned with building an army. Let us assume we have extensive information about past battles, including how long they lasted, how many witches were involved, and how many soldiers were involved. The Queen building that army wishes to know if she should try to find more and better witches or more and better regular soldiers to optimize the speed at which a battle is won. In other words, she wants to know which of these factors has a greater effect on the outcome of the battle.

One solution is to use the classical linear modeling paradigm. A classical linear model has the form

$$\mathbf{Y} = \mathbf{XB} + \mathbf{E}$$

In scalar form, this is

$$Y_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k + \varepsilon_i$$

In this particular example, the dependent variable, $Y_i$, is the time to defeat the enemies in each war fought in the past. The independent variables are $x_1$, the force of the witches, and $x_2$, the force of the regular soldiers. The $\varepsilon_i$ are the model's residuals (a.k.a. the errors). To see how an army would fare, it is reasonable to run two linear models and evaluate the relationship between the number of enemies, the force of the witches, and then the force of the regular soldiers. This being said, when the slopes ($\beta_j$s) are closer to zero or are more negative, then there is a decrease in the number of enemies and the group most reasonable to select to fight in an army.

The classical linear model, especially as fit using ordinary least squares (OLS), is a fantastic way to test relationships between variables. Although it is simple and powerful — and well-established — several requirements must be met to properly use OLS estimation:

1. The residuals are independent; $\mathbb{C}ov\left[\varepsilon_i, \varepsilon_j\right] = 0$ (unless $i = j$)
2. The expected value of the residuals is a constant zero; $\mathbb{E}\left[\varepsilon\right] = 0$
3. The residuals are homoskedastic; $\mathbb{V}\left[\varepsilon\right] = \sigma^2 < \infty$
4. The residuals follow a Normal distribution; $\varepsilon \sim \mathcal{N}(\mu, \sigma)$

All of these requirements can be combined into this one distributional statement as

$$\mathbf{Y} \mid \mathbf{XB} \sim \mathcal{N}\Big(\mathbf{XB};\ \sigma^2\mathbf{I}\Big) \tag{1}$$

Violations of these requirements are easy to detect. Many can be mitigated using a transformation (or more) of the dependent variable. Such transformations include the logarithmic and the logit functions. However, transformations may not be sufficient to fix all of the violations, and they always increase the level of complexity of the model. In the case that the requirements for OLS *are* violated, there are other types of modeling we can use to analyze relationships.

## 1.2 Generalized Linear Models (GLMs)

Assume the CLM model is tested for violations of the requirements and it is found that the residuals do not follow a Normal distribution, or there is nonlinearity in the model. Perhaps the forces of soldiers and witches correspond to an exponential decay in the time until a battle is won. Instead of trying to fix this by performing several transformations and failing, we could instead use a *generalized* linear model (GLM) [For20]. A GLM can be used to model data generated from any distribution in the exponential family, like the binomial or the Poisson or the exponential distribution [Woo06]. The form of a generalized linear model is:

$$\mathbf{Y} = \mathbf{XB} + \mathbf{E}$$

The major difference is that the conditional distribution of $\mathbf{Y}$ is no longer restricted to just the Normal distribution as in equation (1). Any member of the exponential class of families will work (Section 1.2.4). With that being said, this generalization introduces a few complications. Technically, these are

not complications, but rather, they are clarifications of things assumed, but never explicitly examined, in classical linear models.

### 1.2.1 The Linear Predictor

The first complication is the linear predictor. The linear predictor, $\eta = \mathbf{XB}$, can be written in scalar form as

$$\eta_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} \tag{2}$$

This form is identical to the linear predictor in classical linear models. We are still trying to estimate the values of the individual $\beta_j$ using the independent variables $x_{i,j}$.

### 1.2.2 The Conditional Distribution

The second complication is the conditional distribution of the dependent variable. In CLMs, the distribution was the Normal distribution (and *only* the Normal distribution) [For20]. In GLMs, this distriubtion can be *any* member of the exponential family of distributions (Section 1.2.4).

Examples of important distributions in the exponential class include the Normal, the Poisson, the binomial, and the Bernoulli distribution. All four are quite useful in modeling different types of dependent variables.

### 1.2.3 The Link Function

The third, and final, complication is the function that links the linear predictor, $\eta$, with the mean of the conditional distribution, $\mu$. Since the linear predictor

is unbounded, then the the conditional distributions *may be* bounded. The link function effectively unbounds the expected value function [Yee15].

Specifically, Nelder and Wedderburn [NW72, Woo06] defined $g(\cdot)$ to be a link function if

$$g(\mu_i) = \eta_i$$

Here, $\mu_i$ is the expected value of the distribution, conditioned on the values of the independent variables, and $\eta_i$ are the values calculated from the linear predictor.

In the case of the classical linear model, the conditional distribution was the Normal distribution. Its mean is unbounded because $\mu \in (-\infty, \infty)$. That is, the link function used in CLMs is the "identity" function, $\mu_i = \eta_i$.

### 1.2.4 The Exponential Family

Generalized linear models require that the dependent variable's conditional distribution is a member of the exponential family of distributions. The exponential family refers to a set of distributions in which the probability function can be written in the form

$$f(y) = \exp\left[\frac{y\theta - b(\theta)}{a(\phi)} + c(y, \phi)\right] \tag{3}$$

Here, $a$ $b$ and $c$, are arbitrary functions, $\phi$ is the scale parameter, and $\theta$ is the canonical parameter [Woo06]. Before investigating the meaning of these five elements, let us look at an example of a probability mass function written in the form of equation (3)— the Poisson. Below is the proof that the Poisson

belongs to the exponential family.

$$
\begin{aligned}
f(y) &:= \frac{\lambda^y \ \mathbb{e}^{-\lambda}}{y!} \qquad (4) \\
&= \exp\left[\log\left[\ \frac{\lambda^y \ \mathbb{e}^{-\lambda}}{y!}\ \right]\ \right] \\
&= \exp\left[\log \lambda^y \ + \log \mathbb{e}^{-\lambda} - \log y!\ \right] \\
&= \exp\left[\ y \ \log(\lambda) - \lambda - \log y!\ \right] \\
&= \exp\left[\ \frac{y \ \log(\lambda) - \lambda}{1} - \log y!\ \right]
\end{aligned}
$$

And so, if we define

- $\theta = \log(\lambda)$, that is $\lambda = \mathbb{e}^{\theta}$,
- $b(\theta) = \lambda$, that is $b(\theta) = \mathbb{e}^{\theta}$,
- $a(\phi) = 1$, and
- $c(y, \phi) = -\log(y!)$,

then we see that equation (4) can be written in the form of (3); thus, the Poisson is a member of the exponential family of distributions.

Note that we can now interpret the five elements. The scale parameter $a(\phi)$ refers to the level of dispersion of the distribution. In this case, that level is fixed at 1. It can be shown that

$$\mathbb{V}\left[Y_i \mid \mathbf{XB}\right] = a(\phi)\mathbb{V}\left[\mu\right]$$

or, equivalently,

$$a(\phi) = \frac{\mathbb{V}\left[Y_i \mid \mathbf{XB}\right]}{\mathbb{V}\left[\mu\right]}$$

Thus, the dispersion is the ratio of the variances. If the data are conditionally Poisson, then the ratio of the variances will be 1. If they are not, then the ratio of variances will be greater than 1 (overdispersion) or less than 1 (underdispersion). Cases of over- and underdispersion frequently arise when modeling aggregated count data.

The $b(\theta)$ function is used to calculate the expected value and variance of the distribution. It can be shown that $b'(\theta) = \mu$ and that $b''(\theta) = \mathbb{V}\left[\mu\right]$. These facts are central to the simplifications in the maximum likelihood estimation method typically used for GLMs (Section 1.3.1).

The canonical link function is $\theta = \log \lambda$. While canonical links have some useful statistical properties, there is no real reason to limit ourselves to using them. Any function that can map the bounded $\mu$ function to an unbounded $\eta$ function will work.

The $c(y, \phi)$ function has no meaning beyond ensuring the function is a probability function; that is, that it sums (or integrates) to 1 over its domain (sample space).

The GLM can model using only a distribution in the exponential family. So, if the witch and regular soldier data followed an exponential distribution (which *is* a member of the exponential family), then I could use a GLM to model the data and to make predictions.

From the point of a practitioner, the CLM model is not too different from a GLM. We could still use CLM to predict whether witches or regular soldiers should be in the army, but this would be a tedious amount of work.

We know we would transform the dependent variable, check our assumptions again (most likely realize our residuals are overdispersed), then back-transform the variable again to do the analysis. With a GLM, we just run the model with the appropriate distribution and link function, back transform for our predictions and plots, and be good to go. It is faster, cleaner, and easier to do.

### 1.2.5 Generalized Additive Models

As an aside, an interesting extension of a GLM is a generalized *additive* model (GAM). This differs from the GLM in a couple of ways [Yee15]. The form is:

$$g\big(\mu(x_i)\big) = \eta_i = \beta_1 + f_2(x_{i,2}) + \cdots + f_k(x_{i,k})$$

Notice how this is different from the GLM. We are now using **smoothing functions** to transform the independent variables and modeling them. There are a multitude of smoothing functions that are used for different purposes; it is up to the researcher to choose which one best fits their goal. This flexibility allows a researcher to fit data that is not a nice, smooth curve; a GAM can fit data with multiple local minima and maxima, so if we have some wavy data, we can apply a smoothing function and still make a model to estimate beta. For example, assume the data of witches and how many enemies they take down grows, then hits a peak, then falls; perhaps because they get tired after about 50 enemies or so. A GLM could still fit this, but it may be better to try a GAM to smooth out that peak so the model can make estimates more accurately [Yee15].

## 1.3 Estimation Methods

Perhaps more interesting than the model itself is how the model is estimated. How is it possible for us to type a few simple commands and have a model ready at our disposal? It is an interesting algorithm.

This section discusses two types of estimation methods used to fit GLMs and VGLMs. Note there are other ways to do so, but these are two of the most and easiest to use in terms of understanding.

### 1.3.1 Maximum Likelihood Estimation

Maximum likelihood estimation is a method that determines values for the parameters of a model. The parameter values are found such that they maximize the likelihood that the process described by the model produced the observed data [LL22].

This idea is a little complex, so we will review an example. Assume the Queen has won the war and now the kingdom is safe. As Queen of a magical kingdom, she will have information on old battles and when a mystic, ancient foe will decide it wants the kingdom. Suppose we have ten data points representing when the previous battles were. Each of these points represents the amount of time before that battle occurred. The Queen wants to predict when the next battle will come so she can ensure there is a proper armada in place.

First, we need to decide which model best represents her data and which distribution the data best follows. Well, we are concerned with the time between events. The Queen wishes to predict best when the next war will happen, so let us assume that the data follow an exponential distribution. We are aiming to estimate $\lambda$; in other words, we are trying to find an exponential

distribution in which $\lambda$ best fits our data so it is possible to get an accurate prediction of when the next war will be. If there is a graph of multiple curves with different rates, we are trying to find the curve that was most likely responsible for creating the data we already have. Maximum likelihood estimation is a method that will find the value of $\lambda$ that will result in the curve that best fits the data.

**Example**. The above was what maximum likelihood estimation is, and what the goal of the algorithm is. How is it possible to calculate the parameter values, or the **maximum likelihood estimates** (MLEs)? Using the example above, how can one find the MLE of $\lambda$? We need to calculate the total probability of observing all the data; these probabilities combined are known as the joint probability distribution of the observed points [LL22]. Finding the joint probability can be very difficult, considering one will need to use conditional probabilities. Let us make our first assumption that each data point is generated independently of the others. Using this, the total probability of observing all of the data is just the product of observing each individual point (that is, the joint distribution is the product of the marginal distributions). This is defined as:

$$\mathcal{L}(\lambda; \mathbf{x}) = \prod_{i=1}^{n} \lambda \, e^{-\lambda x_i}$$

Notice that this is explicit. The likelihood is a function of the parameter of interest, $\lambda$, given the data.

Calculus is the typical method for calculating the value of $\lambda$ that produces the maximum likelihood. However, since it is usually easier to take the

derivative of a sum instead of a product, one typically maximizes the *logarithm* of the likelihood function, like so:

$$\begin{aligned} l\left(\lambda; \mathbf{x}\right) &:= \log\left(\mathcal{L}(\lambda; \mathbf{x})\right) \\ &= \log\left(\prod_{i=1}^{n} \lambda\ e^{-\lambda x_i}\right) \\ &= \sum_{i=1}^{n} \log\left(\lambda\ e^{-\lambda x_i}\right) \\ &= \sum_{i=1}^{n} \left(\log\lambda\ -\lambda\ x_i\right) \end{aligned}$$

In this case, solving for the parameter, $\lambda$, is a straightforward application of differential calculus, defined below [LL22].

$$\begin{aligned} \frac{d}{d\lambda}\sum_{i=1}^{n} \left(\log\lambda\ -\lambda\ x_i\right) &= \sum_{i=1}^{n} \frac{d}{d\lambda}\left(\log\lambda\ -\lambda\ x_i\right) \\ &= \sum_{i=1}^{n} \left(\frac{1}{\lambda} - x_i\right) \\ &= \left(\frac{n}{\lambda} - n\overline{x}\right) \end{aligned}$$

Setting the derivative to zero and solving for the estimator gives us

$$\begin{aligned} 0 &\overset{\text{set}}{=} \frac{n}{\hat{\lambda}} - n\overline{x} \\ n\overline{x} &= \frac{n}{\hat{\lambda}} \\ \hat{\lambda} &= \frac{1}{\overline{x}} \end{aligned}$$

In other words, the maximum likelihood estimator of $\lambda$ is $\frac{1}{\overline{x}}$.

Maximum likelihood estimation is a fairly reliable method to estimate the parameter(s) of interest. The logarithm of the likelihood function is defined, then maximized, then simplified to the most reasonable value for our parameter. When the likelihood function is known, this is a fairly straightforward calculation.

Although MLE is a reliable method, it relies heavily on the probability mass (or density) function of the distribution rather than the data itself. What happens if the distribution of the data is unknown? Maximum likelihood estimation is not very data-driven; what this means is it utilizes information about the distribution to estimate parameters rather than using the data to estimate the parameters. Is there a way to fix this?

### 1.3.2 Iteratively Reweighted Least Squares (IRLS)

Iteratively Reweighted Least Squares, or IRLS, is an estimation method that uses least squares regression to fit a curve that represents the data. Then, IRLS updates this curve using weights, or the error from the data point to the line, then repeats these steps until the parameters are estimated. The idea seems complex, so allow us to walk through it slowly [Yee15].

Assume we are using the same example above, where the Queen wishes to estimate when the next war is. The goal is to use a GLM, with an exponential family function, to predict when the next war will be. Similar to the above, to fit an accurate GLM, a researcher needs to know the most likely $\lambda$ that represents the data so the Queen can have an accurate (and precise) prediction.

To estimate $\lambda$, the GLM model is first created. Then, the residuals (or errors) become the first weight. A second GLM model is created, using the weight (the error) from the first model. The sum of squares regression (SSR)

is also computed. This calculation is done numerous times. At each step, the new weight is computed and the model is made with the previous weight. Once the total error of the model minus the SSR is near zero, the program is complete and the GLM model is fit, with accurate estimated parameters.

The IRLS algorithm can use MLE, but it tends to be more data-driven as it uses information about the data from the model and creates an improved model based on the flaws of the previous model. The two estimation methods are both quite different; maximum likelihood estimation uses the function of the distribution to create estimates, whereas IRLS creates a model that fits the data and continuously updates it based on errors until the algorithm converges, or reaches the minimum error the user prefers.

The best way to visualize IRLS and its steps is to create an algorithm. Let there be data $X$ and $Y$, where $X = (17, 51, 222, 1, -69)$ and $Y = (100, 856, 701, 2, 11)$. When we create an OLS model, we get an output like so:

| Coefficient | Estimate | p-value |
|---|---|---|
| Intercept | 215.012 | 0.284 |
| $X$ | 2.678 | 0.182 |

The residual standard error is 336.1, with a coefficient of determination of $R^2 = 0.4991$. This model is the starting point of the IRLS algorithm. We will create a new model with weights based on the residuals of the first model. The number of iterations (or the number of times the old model is used to create a new one) is up to the researcher, but let us set the number of iterations to 1,000 or until the sum of the squared residual error is "near zero."

Below is the code that makes the IRLS algorithm for this problem.

```
x <- c(17, 51, 222, 1, -69)
y <- c(100, 856, 701, 2, 11)

mod <- lm(y~x)
summary(mod)

SSR <- sum(resid(mod)^2)

for( i in 1:1000) {
  wt <- sqrt( 1/abs(resid(mod)) )
  mod1 <- lm(y~x, weight=wt)

  if( abs((sum( resid(mod1)^2)-SSR)) < 1e-06) {
    break
  }
  SSR <- sum(resid(mod1)^2)
cat(wt, "\n")
cat(SSR, "\n")
}
```

The summary for the updated model is below.

| Coefficient | Estimate | p-value |
|---|---|---|
| Intercept | 179.105 | 0.1900 |
| $X$ | 2.550 | 0.0815 |

Notice the coefficients and the p-values stayed the same but the residual standard error of the model is now 75.05 with a coefficient of determination of $R^2 = 0.69$. This new model is a large improvement from just the first model.

The IRLS algorithm is a common way to fit a GLM. VGLMs can also be fit using IRLS, but it tends to take a bit longer as there could be different parameters to fit due to the difference in distribution specifications. For exam-

ple, in Section 2, we will use a binomial family to fit a GLM. This calculation takes the user a few seconds in R. When we use a VGLM with a beta-binomial family, the model takes the user multiple minutes. Using IRLS for a VGLM usually takes longer than using it for a GLM because there are more things to be accounted for. For example, different distributions with different parameter specifications that are outside the exponential family, and other factors that make the algorithm take longer.

## 1.4 Vector Generalized Linear Models (VGLMs)

We have seen how powerful the GLM is when data do not conform to the CLM requirements, but it also has limitations. If the conditional distribution of the dependent variable were to fall outside of the exponential family and the requirements could not be fixed to use CLM, how would predictions be made? Another issue is the GLM can only model one linear predictor; in ophthalmology, it is common to see how variables (such as the presence of cataracts, age, etc.) affect the power of both eyes seeing together. The CLM and the GLM could not perform this task.

The vector generalized linear model (VGLM) is very similar to the generalized linear model, but two significant differences set them apart. The VGLM can both help with both of the issues described above; VGLMs can model data that is distributed outside of the exponential family and allow for more than one linear predictor [Yee15].

The matrix form of a VGLM is identical to that of CLM and GLM:

$$\mathbf{Y} = \mathbf{XB} + \mathbf{E}$$

But, the conditional distribution of $\mathbf{Y}$ is not restricted to neither a Normal distribution nor is it restricted to a distribution in the exponential family. For example, in the case of overdispersion, instead of using a GLM model with a binomial distribution fit using quasi-likelihood estimation, we can use the beta-binomial to create a model of overdispersed data.

The linear predictor of a VGLM is similar to equation (2). It is defined below.

$$\eta_{i1} = \beta_{(1)0} + \beta_{(1)1}x_{i,1} + \beta_{(1)2}x_{i,2} + \cdots + \beta_{(1)p}x_{i,p}$$

Notice the change: $\eta$ has a subscript 1, which represents which linear predictor we are expressing. This subscript could also be a 2, or a 3, or however many linear predictors there are in the researcher's model. This is a significant difference between the GLM and VGLM, as the GLM can have only a single linear predictor.

The VGLM still must state the conditional distribution, but this distribution does not need to be in the exponential family. Examples of distributions that can be used for the VGLM are the beta, beta-binomial, beta-prime, and Kumaraswamy distributions [Yee15]. We can also use multi-parameter distributions like the gamma, log gamma, Lomax, and Singh-Maddala [Yee15]. One could still use the gamma distribution for a GLM, but it would be problematic since the GLM would hold one parameter constant.

We have seen that the VGLM can model distributions outside of the exponential family, but how does one model more than one linear predictor? How do we specify which parameters rely on specific independent variables? We will need to use constraint matrices.

### 1.4.1 Constraint Matrices

Constraint matrices can be used for constraining variables or functions in a variety of ways. In this project, constraint matrices are used to relate one or more independent variables (and intercepts) to the proper parameter. For example, assume the data are generated from a Weibull distribution with parameters $\lambda$ and $k$. Three variables relate to the scale, or $\lambda$, and two variables relate to the shape, or $k$. In this case, I would use constraint matrices to specify to the algorithm making the VGLM the first three variables contribute to the scale and the last two contribute to the shape. Constraint matrices are unique for the VGLM and cannot be done in either GLM or CLM; instead, the CLM and GLM model only the expected value.

Constraint matrices have the following properties [Yee15]:

1. the constraint matrices are $M \times R_k$ for $1 \leq R_k \leq M$, where $M$ is the number of linear predictors and $R_k$ is the number of response (independent) variables,
2. the constraint matrices have full column rank, so $(\mathbf{H}_k^T \mathbf{H}_k)^{-1}$ exists,
3. the constraint matrices are known, and
4. they are fixed (not random variables).

The rows of the constraint matrix then become the linear predictors, $n_j$, and the columns are the response variables $\beta_{j,i}$. The entries of a constraint matrix are usually zeroes and ones to denote the presence or absence of a variable in the linear predictor [Yee15]. The first step in finding the constraint matrices is first writing out the $\eta$ matrix, $\mathbf{H}$. Let us use the example of the Weibull distribution above, where $\eta_1$ is determined below by $\lambda$.

$$\eta_1 = \lambda = \beta_{(1)0} + \beta_{(1)1}x_1 + \beta_{(1)2}x_2 + \beta_{(1)3}x_3$$

Then, $\eta_2$ is determined by $k$.

$$\eta_2 = k = \beta_{(2)0} + \beta_{(2)1}x_1 + \beta_{(2)2}x_2$$

Recalling the rows are the $\eta_j$ and the columns are the $\beta_{j,i}$, we can write $\mathbf{H}$ as:

$$\mathbf{H} = \begin{array}{c} \\ \eta_1 \\ \eta_2 \end{array} \begin{array}{c} \begin{array}{ccccccc} \beta_{(1)0} & \beta_{(1)1} & \beta_{(1)2} & \beta_{(1)3} & \beta_{(2)0} & \beta_{(2)1} & \beta_{(2)2} \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \end{array}$$

This matrix does not actually constrain anything quite yet; putting the above matrix into R would produce an error. The next step is to define the constraint matrices. Let us first define the constraint matrix for $\beta_{(1)1}$. The intercept is a bit more complex, so we will address it at the end. The constraint matrix must be full column rank. To make this true for $\eta_1, \beta_{(1)1}$, let us remove the other variables we are not currently interested in:

$$\mathbf{H} = \begin{array}{c} \\ \eta_1 \\ \eta_2 \end{array} \begin{array}{c} \begin{array}{ccccccc} \beta_{(1)0} & \beta_{(1)1} & \beta_{(1)2} & \beta_{(1)3} & \beta_{(2)0} & \beta_{(2)1} & \beta_{(2)2} \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

This is not quite full column rank, so we will now write the constraint matrix for $\eta_1, \beta_{(1)1}$ as:

$$\beta_{(\mathbf{1})\mathbf{1}} = \begin{matrix} \\ \eta_1 \\ \eta_2 \end{matrix} \begin{matrix} \beta_{(1)1} \\ \left[\begin{matrix} 1 \\ 0 \end{matrix}\right. \end{matrix}$$

We can take out the other variables because the constraint matrices are unique for the response variables; we were concerned with creating the constraint for just $\beta_{(1)1}$, so we can place a zero in the other slots we are not constraining just yet. We selected the first variable we wished to constrain— ignoring everything else— to get to full column rank, and found the final matrix. Finding the constraint matrices for the other response variables follows the same steps.

For the intercept, specifically for a two-parameter distribution like the Weibull, they are typically input together. This is because constraining the two intercepts gives us a full column rank matrix. We know that $\eta_1$ has an intercept $\beta_{(1)0}$ and $\eta_2$ has an intercept $\beta_{(2)0}$.

$$\beta_{(\mathbf{1})\mathbf{0}}, \beta_{(\mathbf{2})\mathbf{0}} = \begin{matrix} \\ \eta_1 \\ \eta_2 \end{matrix} \begin{matrix} \beta_{(1)0} & \beta_{(2)0} \\ \left[\begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix}\right. \end{matrix}$$

This matrix is full column rank, constrains the intercepts to the corresponding linear predictors, and is ready to be input into R to create a model. This followed the same steps as above: peel off the parameters we are not interested in, since they are just zeroes, we can drop them, and get the final constraint matrix. Below are all the constraint matrices defined.

$$\beta_{(\mathbf{1})\mathbf{0}}, \beta_{(\mathbf{2})\mathbf{0}} = \begin{matrix} \\ \eta_1 \\ \eta_2 \end{matrix} \begin{matrix} \begin{matrix} \beta_{(1)0} & \beta_{(2)0} \end{matrix} \\ \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \end{matrix}$$

$$\beta_{(\mathbf{1})\mathbf{1}} = \begin{matrix} \\ \eta_1 \\ \eta_2 \end{matrix} \begin{matrix} \begin{matrix} \beta_{(1)1} \end{matrix} \\ \begin{bmatrix} 1 \\ 0 \end{bmatrix} \end{matrix}$$

$$\beta_{(\mathbf{1})\mathbf{2}} = \begin{matrix} \\ \eta_1 \\ \eta_2 \end{matrix} \begin{matrix} \begin{matrix} \beta_{(1)2} \end{matrix} \\ \begin{bmatrix} 1 \\ 0 \end{bmatrix} \end{matrix}$$

$$\beta_{(\mathbf{1})\mathbf{3}} = \begin{matrix} \\ \eta_1 \\ \eta_2 \end{matrix} \begin{matrix} \begin{matrix} \beta_{(1)3} \end{matrix} \\ \begin{bmatrix} 1 \\ 0 \end{bmatrix} \end{matrix}$$

$$\beta_{(\mathbf{2})\mathbf{1}} = \begin{matrix} \\ \eta_1 \\ \eta_2 \end{matrix} \begin{matrix} \begin{matrix} \beta_{(2)1} \end{matrix} \\ \begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{matrix}$$

$$\beta_{(\mathbf{2})\mathbf{2}} = \begin{matrix} \\ \eta_1 \\ \eta_2 \end{matrix} \begin{matrix} \begin{matrix} \beta_{(2)2} \end{matrix} \\ \begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{matrix}$$

If we were to put these in R, then we would make a list of the constraint matrices and input them into the model:

```
HList <- list(
    "(Intercept)" = matrix( c(1,0,0,1), ncol = 2),
    "var11" = rbind(1,0),
    "var12" = rbind(1,0),
    "var13" = rbind(1,0),
    "var21" = rbind(0,1),
    "var22" = rbind(0,1)
    )

modWeibull <- vglm(depVar ~ var11 + var12 + var13 + var21
+ var22, family = weibullR(zero=NULL), constraints = HList)
```

This code then makes a constrained model that uses both parameters to make estimates. The GLM and CLM have limitations when using two-parameter distributions, or when modeling more than one linear predictor. The flexibility of the VGLM with constraints and allowing for multiple $\eta_j$ is quite useful in statistics.

### 1.4.2 Vector Generalized Additive Models (VGAMs)

Like the GLM extended to GAMs, the VGLM extends to vector generalized additive models. The form of a vector generalized *additive* model is below.

$$g(\mu(\mathbf{x}_i)) = \eta_1(\mathbf{x}_i) = \beta_{(1)1} + f_{(1)2}(x_{i,2}) + \cdots + f_{(1)d}(x_{i,d})$$

This model still requires an additive effect between the variables, like in GAMs. Recall that $f(x_i)$ is the smoothing function used on the variable to accurately model data that may have a lot of maxima and minima. There are multitudes

of smoothing functions, so it is up to the researcher to decide which best fits the behavior of their data.

VGAMs also can have more than one linear predictor. If the data have a shape and scale parameter but the distribution seems curvy and has places of increasing and decreasing, smoothing functions can be applied to the variables that belong to the separate linear predictors. The VGAM, like the GAM, will essentially smooth out peaks so the estimates are more accurate (Section 1.2.5).

# 2 Putting Models to Use in R

Now that we have seen what models are available, what their requirements are, and how they differ, we are ready to see a data example. Let us use the fantasy example, but this time, apply it to electoral forensics.

A Queen or King typically rules over more than one kingdom. Instead of going to battle and putting people in danger, let the Queen run against King Darrow. For some background, assume he is known as a brutal King, ruling over Vetraheim (the Winter realm) with an iron fist. He has been the King since the election ten years ago. Many others have run against him, but, somehow, King Darrow always comes out on top. As in most elections, the Queen began her campaign, built alliances, and became popular quite fast. It was going smoothly until the Queen discovered King Darrow doubled his military and was punishing those who were going against him.

When it was time for the election, King Darrow won by over 90% of the votes counted. This percentage is quite an overwhelming loss, so the Queen went to the council, and they believed King Darrow was guilty of electoral fraud. Specifically, they believed he was guilty of differential invalidation. The

| Queen_Num_Votes | King_Darrow_Num_Votes | Num_Valid | Num_Invalid | Total | Prop_Queen_Votes | Prop_Darrow_Votes | Governrate |
|---|---|---|---|---|---|---|---|
| 9,571 | 261,140 | 273,711 | 15,605 | 289,316 | 0.03496753875 | 0.9540719956 | Vetraheim |
| 40,813 | 934,608 | 975,421 | 91,503 | 1,066,924 | 0.04184142027 | 0.9581585797 | Sumarheim |
| 39,833 | 1,270,115 | 1,309,948 | 134,641 | 1,444,589 | 0.03040807727 | 0.9695919227 | Varaheim |
| 5,669 | 237,151 | 242,820 | 10,288 | 253,108 | 0.02334651182 | 0.9766534882 | Haustenheim |
| 9,696 | 296,960 | 306,656 | 28,060 | 334,716 | 0.03161849108 | 0.9683815089 | Veorriki |
| 6,045 | 114,332 | 120,377 | 15,091 | 135,468 | 0.05021723419 | 0.9497827658 | Jorpriki |
| 43,817 | 1,569,113 | 1,612,930 | 116,727 | 1,729,657 | 0.02716608904 | 0.972833911 | Bruniriki |
| 50,239 | 1,406,935 | 1,457,174 | 138,385 | 1,595,559 | 0.03447700824 | 0.9655229918 | Vatnriki |
| 1,919 | 151,331 | 153,250 | 3,810 | 157,060 | 0.01252202284 | 0.9874779772 | Myrkriki |
| 36,981 | 1,783,362 | 1,820,343 | 97,809 | 1,918,152 | 0.0203154021 | 0.9796845979 | Kyndaheim |
| 6,961 | 145,354 | 152,315 | 24,625 | 176,940 | 0.04570134261 | 0.9542986574 | Gleymaheim |
| 46,775 | 1,786,602 | 1,833,377 | 117,112 | 1,950,489 | 0.02551302869 | 0.9744869713 | Hjooariki |
| 38,948 | 1,486,429 | 1,525,377 | 113,576 | 1,638,953 | 0.02553335995 | 0.9744666401 | Ulfrheim |
| 25,883 | 732,609 | 758,492 | 63,180 | 821,672 | 0.03412428872 | 0.9658757113 | Tysdagar |
| 76,831 | 2,255,243 | 2,332,074 | 247,956 | 2,580,030 | 0.03294535251 | 0.9670546475 | Engillriki |
| 34,061 | 1,175,551 | 1,209,612 | 104,749 | 1,314,361 | 0.02815861615 | 0.9718413838 | Laknariheim |
| 20,260 | 1,128,889 | 1,149,149 | 58,243 | 1,207,392 | 0.01763043783 | 0.9823695622 | Dauoienheim |
| 46,929 | 1,228,458 | 1,275,387 | 101,071 | 1,376,458 | 0.03679589019 | 0.9632041098 | Lysandraheim |
| 4,117 | 85,866 | 89,983 | 9,141 | 99,124 | 0.04575308669 | 0.9542469133 | Old Coore |
| 27,681 | 727,895 | 755,576 | 58,559 | 814,135 | 0.03663562633 | 0.9633643737 | New Coore |
| 6,816 | 199,679 | 206,495 | 37,049 | 243,544 | 0.03300806315 | 0.9669919369 | Port Selja |
| 2,054 | 45,066 | 47,120 | 5,402 | 52,522 | 0.04359083192 | 0.9564091681 | Strondheim |

**Figure 1:** *The data table for the election model.*

more votes he received, the lower his invalidation rate. The council collected the number of ballots cast for King Darrow and the Queen, the number of valid votes, and the number of invalid votes. Because we want to see how his invalidation rate is affected by the more valid votes he has, the independent variable is the proportion of valid votes King Darrow had and the dependent variable is the proportion of invalid votes. Let us evaluate the three models discussed above to see if there is evidence for differential invalidation against King Darrow.

## 2.1 The Data

A few lines of the data used for this example are provided in Figure 1.

## 2.2 The CLM Model

The first step is to check the four assumptions of the residuals. To do so, one must make a model for the dependent versus the independent variable, find

the residuals for that model, and apply visual as well as numerical tests to the residuals.

### 2.2.1 The Model

I am using R for the modeling. After loading the data, I created my independent and dependent variables. Since we are trying to determine if there is a relationship between the invalidation rate and King Darrow's votes, our independent variable is the number of invalid votes divided by the total number of votes. In R, the code will be written below.

```
pInv <- Invalid / (Valid + Invalid)
```

Our dependent variable is the proportion of votes for King Darrow over the total number of votes since we are trying to see changes based on King Darrow's votes and the invalidation rate. The code is a similar notation in R:

```
pDarrow <- King_Darrow_Votes / Valid
```

Now, we are ready to make our linear model, get our residuals, and begin our assumption tests. Below is the model, and how to access the residuals.

```
modA <- lm(pInv ~ pDarrow)
eA   <- residuals(modA)
```

Now, we can perform our assumption tests.

### 2.2.2 Constant Expected Value

The first requirement tested is the constant expected value and the independence of the residuals.

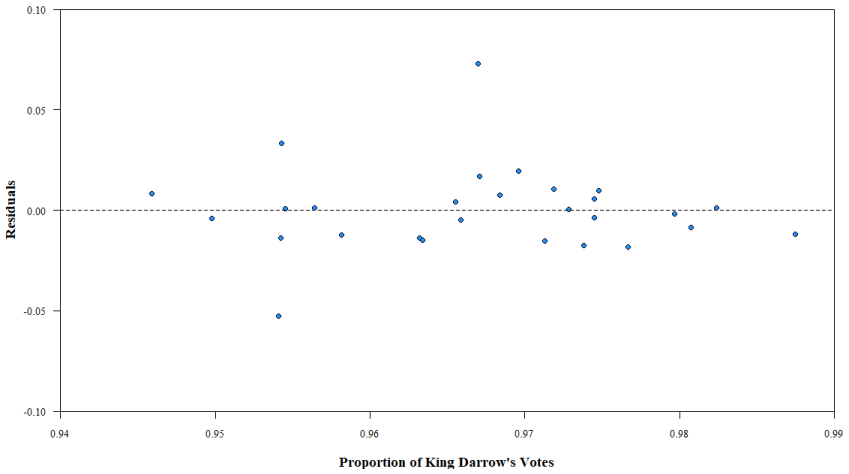


**Figure 2:** *A residuals plot of the CLM election model.*

The numeric test for seeing if there is a constant expected value is the Wald–Wolfowitz runs test, created by Abraham Wald and Jacob Wolfowitz [Bra68]. This test aims to check that elements are mutually independent. The null hypothesis is that the expected value of the residuals is constant and zero. When we perform the runs test, our p-value is 0.700, meaning we have sufficient evidence that the expected value of the residuals is constant and zero and that the residuals are independent.

Had the null hypothesis been rejected, we would need to examine the residual plot to determine if the violation is significant enough to affect the estimates. Figure 2 shows the relationship between the residuals (vertical axis) and the independent variable (horizontal axis).

There isn't a pattern here; it slopes down a little, but if there was an obvious quadratic shape, it would be a bit more worrisome.

### 2.2.3 Constant Variance (Homoskedasticity)

The next requirement we will test for is constant variance (also known as homoskedasticity).

The typical numeric test for homoskedasticity is the Breusch–Pagan test, created by Trevor Breusch and Adrian Pagan in 1979. It tests whether or not the residuals of a model are dependent on the independent variable [BP79]. The null hypothesis is that the residuals do not depend on the independent variable and are therefore homoskedastic.

When performing the Breusch-Pagan test on the model, we get a p-value of 0.450, meaning we cannot reject the null hypothesis that the residuals have constant variance. That is, the model passes this test.

Had we rejected the null hypothesis of homoskedasticity and found a violation of one of the OLS requirements, we would examine the residuals plot to determine if the violation is significant enough to affect our estimates of the standard errors. This graphic allows one to see if there are any patterns, suggesting that the data are heteroskedastic. These patterns are pretty obvious; the shape of the graph will have a funnel, a trumpet, or a balloon shape, in which the data essentially puff out along the y-axis in the center.

Looking at Figure 2, one sees no significant pattern in the spread. There are a few points that are above where $y = 0$, but things seem to fall randomly into place as one would expect. It would be rather obvious if there were a pattern.

### 2.2.4 Normality of Residuals

The final requirement that needs to be checked is the Normality of the residuals. A typical numeric test for normality is called the Shapiro-Wilk test, named

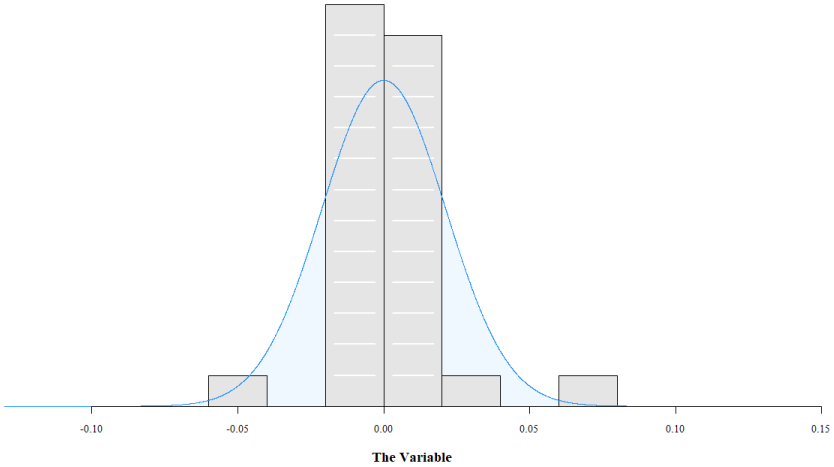


**Figure 3:** *A histogram of the residuals from the model with a Normal distribution provided to illustrate the deviation from Normality.*

after Samuel Sanford Shapiro and Martin Wilk in 1965 [For20, SW65]. The null hypothesis is that the residuals are generated from a Normal distribution. If the p-value is below 0.05, then we reject the null hypothesis and conclude the residuals are *not* from a normal distribution.

For this model, the Shapiro-Wilk test indicates that the residuals are not generated from a Normal distribution (p-value $= 0.003$). Thus, we have a violation of the requirements of OLS.

To determine if the violation is severe enough to significantly affect the estimated confidence intervals, one should examine a histogram with an overlay to show what the Normally distributed residuals should look like. Figure 3 is a histogram with the overlay of the residuals.

The assumptions for the residuals seem to perform well, although there are some violated assumptions. It is possible for some residuals to be outliers due to the distribution of the data.

### 2.2.5 A Second Model

Overall, the OLS model without any transformations seems to perform well; the only issue was with the normality of the residuals. When the sample size is large, perhaps around $n = 100$, then we can use the Central Limit Theorem and assume that our data are normal.

In this case, the sample size is $n = 28$, since there are 28 kingdoms in the land. Let us apply a transformation to our independent variable and fix the violation of normality.

To select an appropriate transform, one must consider the boundedness of the dependent variable. The invalidation rate, being a proportion, is bounded between 0 and 1. Thus, a logit transform is appropriate.

Let us perform another Shapiro-Wilk test on this transformed model to see if the normality violation is fixed. We get a p-value of 0.251; therefore, there is sufficient evidence the residuals are now normal. We need to check our other requirements. Since checking the requirements was done in detail above, below is a table of the remaining requirements.

| Test | p-value |
|---|---|
| Constant Expected Value | 0.4411 |
| Constant Variance | 0.9368 |
| Normality | 0.2510 |

It appears all of our requirements are met, and there are no violations. We can now use the logit-transformed model to perform our testing. With this new model, let us see if King Darrow is guilty of differential invalidation. Below is a summary of the model:

| | Estimate | p-value |
|---|---|---|
| Intercept | 28.629 | $1.130 \times 10^{-5}$ |
| Darrow Support | -31.848 | $3.200 \times 10^{-6}$ |

Since we used a logit transform, the slope should be interpreted as the logarithm of the increase in the *odds* of a ballot being invalidated. In general, we can interpret the slope effect as every 1 increase in the independent variable, the odds of the dependent variable increase by a factor of $e^{\beta}$. So, with our model, that would be an odds change to $1.47 \times 10^{-14}$ of its previous value.

More meaningfully, if the support for King Darrow in the kingdom is increased by just 10%, the odds of a ballot being invalidated decreases by 96% ($= 1 - e^{-31.848 \times 0.10} = 1 - 0.04$). This change is a rather large effect. Figure 4 illustrates this effect (light purple curve).

Most importantly, notice that the slope is negative, and the p-value is less than 0.05. These two things indicate that there is sufficient evidence that there is differential invalidation that *favors* King Darrow in this election.

Notice the purple line is our original OLS model without any transformations; this line is quite obviously linear. Although it fits the data, it is not really an accurate model due to the assumption violations. The green curve is the OLS model with a logit transformation; notice how much nicer this curve fits the behavior of the data. Furthermore, the assumptions are met, so we can use this model for predictions. The transformed OLS model does a great job, but it is quite complex and monotonous to transform, check assumptions, then back-transform.

Let us now look at the generalized linear model and the differences from the classical linear model.

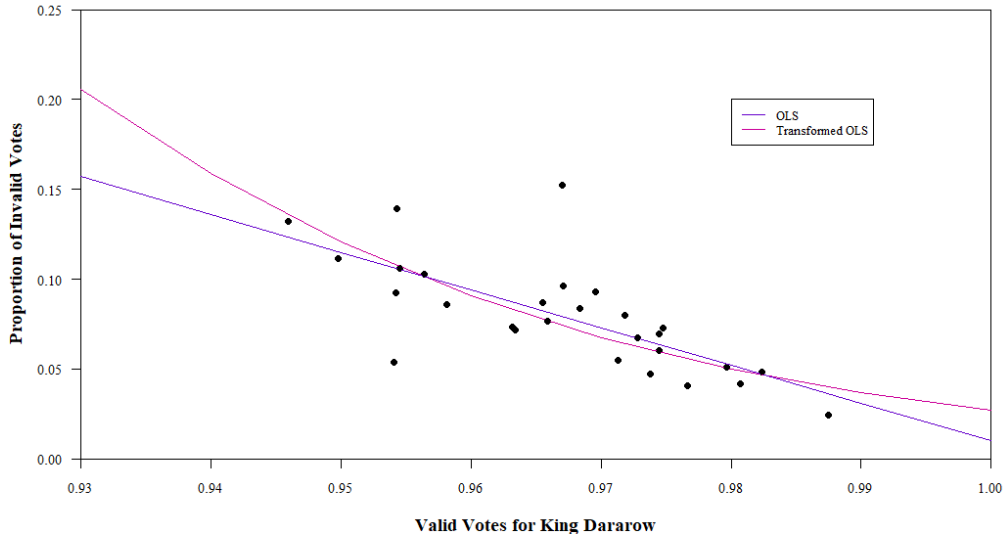


**Figure 4:** *A graphic of both classical linear models. This shows the difference between the transformed model and the non-transformed model.*

## 2.3 GLM Model

Recall the GLM is more flexible than the CLM. It allows us to select certain features to better reflect what we know about the response variable; we just need to know the link function and to model data within the exponential family [Woo06].

Think about the distribution of the data and if that distribution is a member of the exponential family of distributions. The dependent variable is the proportion of invalid votes, and we are aiming to see if there is a relationship between the proportion of invalid votes (invalidation rate) and the proportion of votes for King Darrow. Thus, we are looking for the probability of obtaining a given number of successes (invalid votes, $y$) over the total number of trials (ballots cast, $n$). Written in this way, the distribution of the data is clear and the number of invalid ballots may follow a binomial distribution. Since many distributions can model the number of successes out of the number

of trials, we cannot be certain that the conditional distribution is exactly the binomial. However, it is a good place to start.

One benefit to using the binomial is that it is a member of the exponential family. Here, in much the same way as I did for the Poisson in Section 1.2.4, I show that the binomial distribution is also a member of the exponential family. Recall the exponential form is:

$$f(x) = \exp\left[\frac{x\theta - b(\theta)}{a(\phi)} + c(x, \phi)\right]$$

The probability mass function (pmf) for the binomial distribution is:

$$f(x) = \binom{n}{x} \pi^x \ (1-\pi)^{n-x}$$

Let us continue as in the Poisson example (Section 1.2.4).

$$\begin{aligned} f(x) &= \binom{n}{x} \pi^x \ (1-\pi)^{n-x} \\ &= \exp\left[ \ \log\left(\binom{n}{x} \pi^x \ (1-\pi)^{n-x}\right) \ \right] \\ &= \exp\left[ \ \log\binom{n}{x} + x \ \log\pi + (n-x)\log(1-\pi) \ \right] \end{aligned}$$

The above simplifies to:

$$f(x) = \exp\left[x\big(\log(\pi) - \log(1-\pi)\big) + n\log(1-\pi) + \log\binom{n}{x}\right]$$

Next, let's apply our logarithmic rules and rewrite this as:

$$f(x) = \exp\left[\frac{x\log\left(\frac{\pi}{(1-\pi)}\right)}{1} + n\log(1-\pi) + \log\binom{n}{x}\right]$$

Defining the following shows that this distribution is in exponential family form (3):

- $\theta = \log\left(\frac{\pi}{1-\pi}\right)$, that is $\pi = \frac{\mathbb{e}^{\theta}}{1+e^{\theta}}$,
- $b(\theta) = n\log\left(1+\mathbb{e}^{\theta}\right)$,
- $a(\phi) = 1$, and
- $c(x,\phi) = \log\binom{n}{x}$,

Here, I have shown that the binomial is a member of the exponential class of distributions.

For GLMs, the second thing to specify is a link function. From the proof that the binomial is exponential class, we see that the canonical link for the binomial distribution is the logit function, equation (5),

$$\operatorname{logit}\pi := \log\left(\frac{\pi}{1-\pi}\right) \tag{5}$$

with its inverse, the logistic is:

$$\operatorname{logistic} x := \frac{\mathbb{e}^{x}}{1+\mathbb{e}^{x}}$$

Now, we have all of our requirements covered and we are ready to model. In R, the function to perform generalized linear modeling is `glm`. The entire line of code is:

```
modC <- glm(pInv ~ PDarrow, family=binomial(link = "logit"))
```

The resulting output looks like

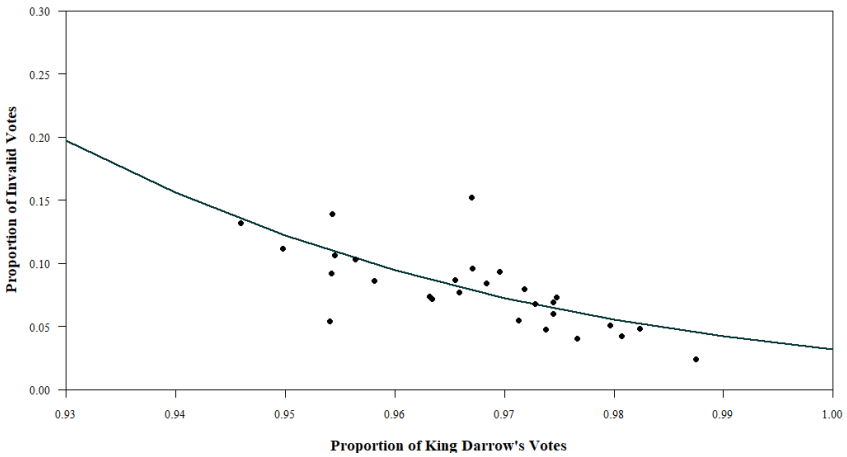


**Figure 5:** *Graphic of the GLM model using a binomial family and logit link. Notice this curve is similar to the transformed OLS model.*

| | Estimate | P-value |
|---|---|---|
| Intercept | 25.226 | $\ll 2.00 \times 10^{-16}$ |
| Darrow Support | -28.634 | $\ll 2.00 \times 10^{-16}$ |

Once again, there is a negative slope indicating that the more votes King Darrow receives, the lower the invalidation rate. Note that the p-value is much less than the usual $\alpha = 0.05$, meaning that there is significant evidence that the differential invalidation aided King Darrow.

One thing to notice here is how much shorter this section is as compared to the OLS section; there was no need to do nearly as many steps or transformations. All that needed to be done was figure out the likely distribution of the dependent variable, decide on an appropriate link function, and run the model. Figure 5 illustrates the effect of support for the King on the invalidation rate according to the GLM model.

While we are here, let us notice some similarities and differences between the CLM models and our GLM model (Figure 6). The regular OLS

curve is linear, so it does not fit the curvature of the data as well as a transformed model or the GLM. Usually, we can get away with the violation of Normality when we have a large enough sample size (this is just the Central Limit Theorem). The logit-transform of our CLM model fits rather well, but it does start a bit steep. The GLM does a bit better than the transformed OLS since the curve starts lower and looks like it fits the data slightly more precisely than the previous model.

All of this aside, the transformed CLM and the GLM curves look almost the same. The most notable difference is the time it took to fit the models because with the GLM we did not have to worry about transformations, checking requirements, and then back-transforming. We just ran our model and did one back-transform to get our curve. That one advantage to using the GLM over the transformed CLM is simply time; doing the GLM model is much shorter than the CLM because there is no need for assumption checking.

Plus, we do end up getting a slightly more accurate model since we can take advantage of the actual conditional distribution of the dependent variable as well as the link function. To drive home this point, Figure 6 is a graphic of the CLM, the transformed CLM, and the GLM model.

So, to recap: The CLM was okay, but it was too linear for our data. While the transformed CLM was better, it was relatively difficult to perform and took longer to do. The GLM was even better, but one issue remains.

Note that the ballots are being aggregated over multiple kingdoms. Because we are working with aggregate data, the model is most likely overdispersed. This is an important observation because the dispersion parameter for the binomial distribution, $a(\phi) = 1$. If that parameter were a variable (as in the case of the Normal distribution), then dispersion would be modeled by

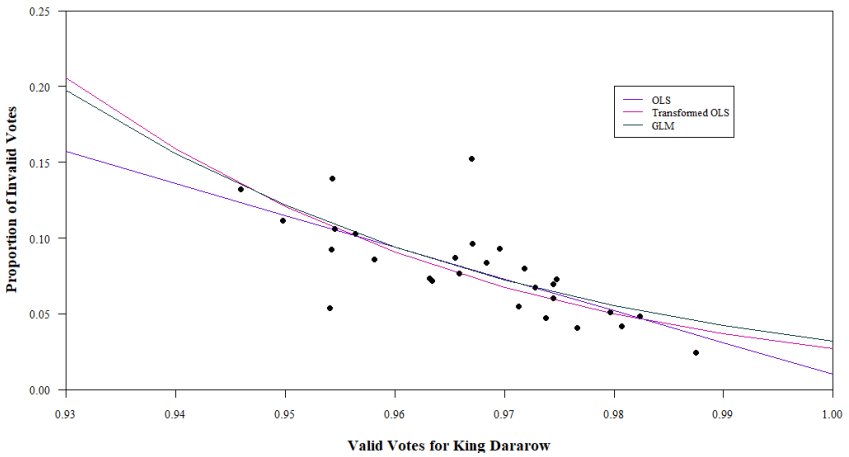


**Figure 6:** *A plot of the GLM model with the two OLS models. Notice how well it fits the data, and the similarities and differences from the OLS models.*

that parameter. However, this is not the case for the binomial (or for the Poisson).

One way to check for overdispersion is to calculate the ratio of the residual deviance to the degrees of freedom. If it is much greater than one, then there is overdispersion. One can also obtain a confidence interval for the residual deviance under the assumption that there is no dispersion.[For20, §12.3.2]

Using the Chi-squared distribution, we are 95% confident that the residual deviance would be between 13.8 and 41.9. Because the observed residual deviance is much greater than this interval, we should conclude that there is evidence of overdispersion in the model.

Now that we know we have overdispersion, how do we find flexibility for overdispersion so there is a more accurate model before we accuse King Darrow of electoral fraud?

## 2.4 VGLM Model

To fix overdispersion in the model, we can use a distribution that is similar to the binomial but does not have the dispersion parameter being $a(\phi) = 1$. There are several, but a natural one is the beta-binomial distribution. This distribution is very similar to the binomial. The main difference between the binomial and the beta-binomial is that the beta-binomial allows the success probability to vary according to a beta distribution.

Its probability mass function is:

$$f(x;\ n, \alpha, \beta) = \binom{n}{x} \frac{B(x+\alpha, n-x+\beta)}{B(\alpha, \beta)}$$

In this function, like the binomial, $x$ is the number of successes and $n$ is the number of trials. In addition, $\alpha$ and $\beta$ are parameters describing the variability of the success probability ($\pi$ in the binomial model). Finally, $B(\cdot, \cdot)$ is the beta function. Using the beta function allows the distribution to account for overdispersion in the model. This distribution, therefore, makes the model more reliable.

The beta-binomial distribution is not a member of the exponential family. Thus, we cannot use the GLM framework. We will have to use a VGLM model instead [Yee15].

With that being said, our coding syntax will not look much different than that of the GLM:

```
modD <- vglm(pInv ~ PDarrow, family=betabinomial)
```

The link function for the beta-binomial defaults to the logit function, so there is no need to specify it. The summary of our model is:

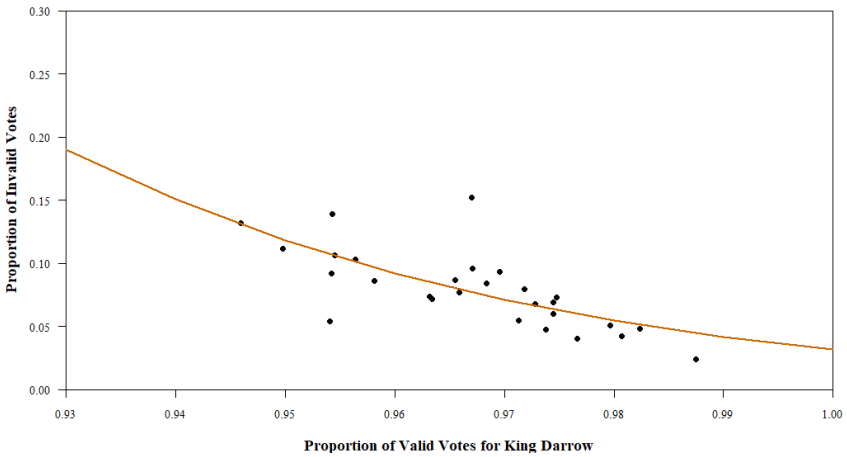


**Figure 7:** *Graphic of the VGLM model. This fits the data the best, but is still similar to the GLM and transformed OLS model.*

| | Estimate | P-value |
|---|---|---|
| Intercept | 24.654 | $4.300 \times 10^{-7}$ |
| Darrow Support | -28.059 | $2.830 \times 10^{-8}$ |

Again, note that the slope is negative — and the p-value is much less than $\alpha$ — so there is evidence of differential invalidation that aided King Darrow.

Notice these results are quite similar to the GLM model. The p-values are different, but the curve matches rather well. The difference is we were able to have flexibility the possibility of overdispersion, which makes for a more accurate model.

Now, the Queen can be confident that everything is perfect — or at least more perfect — before presenting the results to the Court. The downfall of the VGLM is the computational expense. Because there are more parameters to be fit, the IRLS iterations tend to take a bit longer. The time spent running the model is worth it to have a more precise estimate. Figure 7 is a graphic of the VGLM model.

## 2.5 Reflection

We started with the classical linear model, fit using OLS. After modeling using OLS, there were violations of the requirements. If the sample size were larger, one could have felt comfortable relying on the Central Limit Theorem and using OLS. However, the sample size was only 28. Plus, regular OLS was too linear to fit the data well.

Then, time was spent transforming the dependent variable using the logit function. We checked our assumptions again, saw everything was good, and then back-transformed the estimates. This fix made a rather precise model, but it was not perfect and took a lot of time and brain power since we had to keep checking the requirements.

We then took advantage of what we knew about the data and fit everything using a GLM. This approach gave a really accurate model and took hardly any time. This worked well until we realized we could have overdispersion because we were aggregating so many ballots at the kingdom level.

To compensate for overdispersion, we then fit a new type of distribution to allow for overdispersion and used a VGLM instead since the distribution we decided on was outside of the exponential family. This model gave an even more precise model but took quite a bit of computer power.

Overall, it is important to realize *when* to use each type of model. One can use transformed OLS if there is a lot known about it and not so much about other distributions and GLMs; or, one can use a GLM to make things easier if there is confidence about distributions in the exponential family and link functions. Then, one can use a VGLM for modeling what cannot be done using OLS or a GLM to make a really precise model. It is hard to go wrong either way. We talked a lot about differences, so just to show how similar these

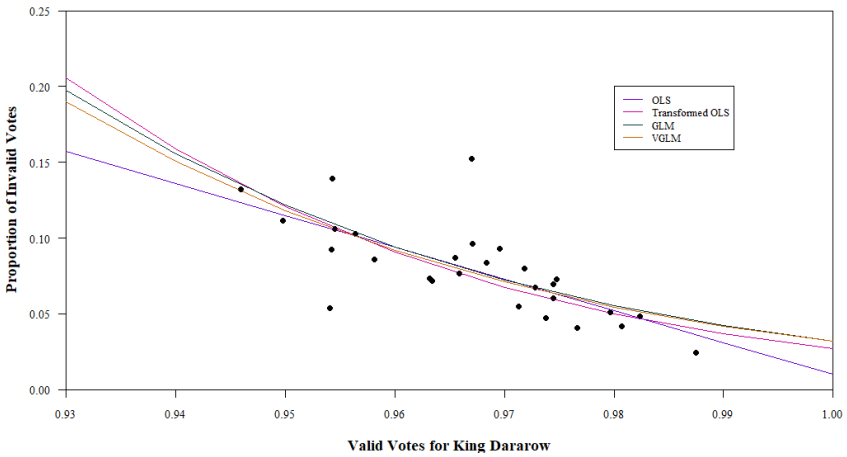


**Figure 8:** *Graphic of all the models together. Notice how similar each model is.*

can be, Figure 8 is a graphic of all the models plotted together. Note that all models, except for the original line, make similar predictions.

Notice how close the GLM and the VGLM models are to each other. They both start at a slightly different place and end up a little different, which could be due to the VGLM correcting for overdispersion and using a different family. The first OLS without any transformations was clearly suboptimal. The transformed OLS seems acceptable, but still a bit off and took a lot of physical time. The GLM worked well, minus the chance of overdispersion. Computationally, the VGLM was time-consuming, but worth it so we can have an accurate model to bring thorough evidence to the Court.

We have just explored one of the two differences a VGLM has from a GLM. Using a VGLM, one can model data outside of the exponential family, like the beta-binomial, the Cauchy, and many more. This is very helpful for overdispersed data. Now, we are ready to look at the other difference. The VGLM can model more than one linear predictor. Recall the form of the VGLM and the $i$ subscript connected to the $\eta$. This means I can have a dependent variable that is modeled by a distribution with two parameters,

where one linear predictor corresponds to the first parameter, and the other can correspond to the second. It is important to mention there can be more than two linear predictors for a two-parameter distribution. Let us delve into an example of a two-parameter distribution.

# 3 Lengthy Application to the Gamma

The purpose of this section is to illustrate the second key difference between the VGLM compared to the GLM; the VGLM can model more than one linear predictor, whereas the GLM can just model one. An example of a distribution with more than one parameter is the Gamma distribution. In this section, we will use $\kappa$ and $\theta$ to illustrate differences in the GLM and VGLM.

## 3.1 Gamma Distribution

The gamma distribution is a two-parameter distribution that is typically used for modeling the time between events, or the time until events occur. The gamma distribution is like an extension of the exponential distribution; the exponential distribution can model time until the next event, so a single instance. The gamma distribution models the time *between* events, or until the next set of events occur. The gamma distribution models multiple instances:

$$f(x;\kappa,\theta) = \frac{1}{\theta^{\kappa}\Gamma(\kappa)} x^{\kappa-1} \mathrm{e}^{-x/\theta}$$

Note that $\kappa$ is the shape and $\theta$ is the scale. The gamma distribution has a mean of $\kappa\theta$ and a variance of $\kappa\theta^2$. What makes the gamma distribution interesting when modeling, especially when using CLM and GLM, is the fact it is a two-parameter distribution. How does each of the models estimate the

| A | B | C | D | E |
|---|---|---|---|---|
| number_witches | number_soldiers | level_witches | level_soldiers | time_until_end |
| 109 | 693 | 7 | 1 | 14 |
| 88 | 728 | 3 | 0 | 26 |
| 100 | 699 | 5 | 1 | 17 |
| 89 | 701 | 7 | 0 | 15 |
| 91 | 695 | 3 | 0 | 28 |
| 103 | 700 | 5 | 3 | 16 |
| 106 | 691 | 6 | 1 | 15 |
| 93 | 714 | 5 | 2 | 16 |
| 96 | 710 | 7 | 2 | 13 |
| 111 | 690 | 4 | 0 | 23 |
| 88 | 678 | 9 | 1 | 12 |
| 94 | 706 | 5 | 2 | 16 |
| 94 | 707 | 6 | 0 | 16 |
| 94 | 728 | 6 | 2 | 14 |
| 101 | 713 | 3 | 1 | 24 |
| 117 | 696 | 7 | 1 | 13 |
| 96 | 704 | 4 | 0 | 22 |
| 108 | 708 | 5 | 0 | 19 |
| 101 | 684 | 7 | 0 | 14 |
| 112 | 699 | 6 | 1 | 15 |
| 112 | 698 | 5 | 1 | 17 |
| 100 | 690 | 2 | 0 | 35 |
| 90 | 713 | 4 | 0 | 24 |

**Figure 9:** *Some of the data used for the gamma model.*

parameters for the gamma distribution, especially when the models handle only one parameter?

## 3.2 Types of Data that may be Gamma-Distributed

Many types of data can be gamma-distributed. To keep consistent with the kingdom example, assume the Queen has data on how many witches and soldiers were in a battle, along with their experience level, and how many days until the battle was complete. She wants to know for the next battle if she wants to recruit more witches or more soldiers. In this case, because we are using the information to determine how fast the battles were over, the

variables become:

$$1/\text{time} \sim \text{Gamma}(\kappa, \theta)$$

Where $\kappa$ and $\theta$ are defined as:

$$\kappa = \kappa_0 + \kappa_1 \text{Number of witches} + \kappa_2 \text{Number of soldiers} + \kappa_\varepsilon$$
$$\theta = \theta_0 + \theta_1 \text{Level of witches} + \theta_2 \text{Level of soldiers} + \theta_\varepsilon$$

To generate the data, R will be used. The number of witches, the number of soldiers, the level of the witches, and the level of the soldiers will be generated from a binomial distribution with different sample sizes and probabilities. The error for the parameters will be generated from a normal distribution. Values for the $\theta_i$s and $\kappa_i$s (or the slopes for each variable) will also be specified. Finally, our dependent variable, time, will be generated from a gamma distribution with the parameters we created above.

Below is the R code to generate the data. See Figure 9 for the data table:

```
n = 1e4

### Kappa
k0 <- log(2.00)
k1 <- log(0.99)
k2 <- log(0.97)

wn   <- rbinom(n,  20, 0.2)
sn   <- rbinom(n, 100, 0.7)
epsk <- rnorm (n,   0, 1e-2)

kappa = k0 + k1*wn + k2*sn + epsk
eKap = exp(kappa)
```

```
### Theta
t0 <- log(3.00)
t1 <- log(0.95)
t2 <- log(0.93)

wl   <- rbinom(n, 10, 0.5)
sl   <- rbinom(n, 10, 0.1)
epst <- rnorm (n,  0, 1e-2)

theta = t0 + t1*wl + t2*sl + epst
eThe = exp(theta)

### Independent variable
depVar = rgamma(n, shape=eKap, scale=eThe)
time = 1/depVar
```

Now that the data are generated, let us look at how the CLM, the GLM, and the VGLM predict the parameters. Below is a table of the actual parameter for the data; it will be important to keep these values in mind when comparing across models.

| Variable | Kappa | Theta |
|---|---|---|
| Kappa Intercept | 0.693 | 0.000 |
| Theta Intercept | 0.000 | 1.099 |
| Witch Level | 0.000 | -0.051 |
| Soldier Level | 0.000 | -0.073 |
| Number of Witches | -0.010 | 0.000 |
| Number of Soldiers | -0.030 | 0.000 |

Then, the equations are:

$$\kappa = 0.693 - 0.010 \text{ Number of Witches} - 0.030 \text{ Number of Soldiers}$$

$$\theta = 1.099 - 0.051 \text{ Witch Level} - 0.073 \text{ Soldier Level}$$

## 3.3 CLM and the Gamma Distribution

The gamma distribution is a two-parameter distribution. So, how does the CLM estimate these parameters? Recall the form of the CLM:

$$Y_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_i x_i + \varepsilon_i$$

The form allows only for one linear predictor; in other words, the CLM can estimate only one parameter. In R, the way the model estimates the two parameters of the gamma distribution is by assuming an additive predictor for all of the variables. In other words, there is no way to specify certain variables are for the $\kappa$ parameter and others are for the $\theta$ parameter. Because of this, the requirements of the CLM will typically be violated, and it will be difficult to transform the independent variable so that all of the requirements are met.

The CLM will be done in R using OLS. When the model is created, the estimations for the variables are:

| Variable | Effect |
|---|---|
| Intercept | $3.17 \times 10^{-7}$ |
| Witch Level | 0.229 |
| Soldier Level | 5.864 |
| Number of Witches | 2.267 |
| Number of Soldiers | 1.252 |

There are a few things to note. There is no way we can separate the variables into the respective parameters. This is because of the additive effect between them, as well as the fact the CLM does not model more than one linear predictor (it models only the expected value). There is no way for the model to specify which parameters have which variables, so there cannot possibly be an estimation of both parameters.

Furthermore, the estimations are not close to what they should be. This is an important limitation of the CLM; in a lot of real-life data, it is not uncommon for there to be more than one linear predictor. Keeping this model in mind, let us determine if specifying a family and link function for the GLM model yields a better result.

## 3.4 Gamma and GLM

Similar to the case of the CLM, the GLM is also limited to one linear predictor. Recall the form of a GLM is:

$$\eta_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}$$

The GLM may handle the gamma-distributed data better since the GLM is built off of distributions that belong to the exponential family; but, there is no way to specify which independent variables belong to the corresponding parameters. The GLM will also assume an additive effect between variables rather than determining the $\kappa$ and $\theta$ parameters. Below, are the estimates for the variables:

| Variable | Effect |
|---|---|
| Intercept | -0.396 |
| Witch Level | 8.547 |
| Soldier Level | 6.225 |
| Number of Witches | 17.241 |
| Number of Soldiers | 19.231 |

Similar to the CLM model, the GLM cannot be used to model two parameters separately. Even using the gamma family and the corresponding link function (the inverse link function in this case, since the dependent variable is inverse time), the model still failed to perform well. It is quite hard to compare the CLM and GLM model to the original parameters for the data because both the CLM and GLM fail to estimate more than one parameter properly. If both models could estimate more than one linear predictor, and if there were a way to specify which variables are in the $\kappa$ and which are in the $\theta$, then perhaps the model could perform better.

## 3.5 Gamma and VGLM

Luckily, the VGLM allows for more than one linear predictor. If we think back to Section 1.4.1, there is also a way to specify which variables belong to the parameters. We are using a two-parameter gamma distribution. Let $\kappa = \eta_1$ and $\theta = \eta_2$. The $\mathbf{H}$ matrix is then defined as:

$$\mathbf{H} = \begin{array}{c} \\ \eta_1 \\ \eta_2 \end{array} \begin{array}{c} \begin{array}{cccccc} \beta_{(1)0} & \beta_{(1)1} & \beta_{(1)2} & \beta_{(2)0} & \beta_{(2)1} & \beta_{(2)2} \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \end{array}$$

The constraint matrix specifies which variables essentially belong to the $\kappa$ and $\theta$ parameters. This model is a little different than the example with the Weibull distribution because of the family function that is used. The family `GammaR` will be utilized; this is a special parameterization for the two-parameter gamma distribution. The output gives $\theta$ first, and then $\kappa$. This being said, the constraints for the variables will need to be flipped. The easiest way to do this is to swap the rows for $\eta_1$ and $\eta_2$. Let us first define the matrix for $\beta_{(1)1}$:

$$\beta_{(\mathbf{1})\mathbf{1}} = \begin{array}{c} \\ \eta_2 \\ \eta_1 \end{array} \begin{array}{c} \begin{array}{cccccc} \beta_{(1)0} & \beta_{(1)1} & \beta_{(1)2} & \beta_{(2)0} & \beta_{(2)1} & \beta_{(2)2} \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Recall the constraint matrices are full column rank. Thus, we can simplify the constraint matrix for $\beta_{(1)1}$ as:

$$\beta_{(\mathbf{1})\mathbf{1}} = \begin{array}{c} \\ \eta_2 \\ \eta_1 \end{array} \begin{array}{c} \begin{array}{c} \beta_{(1)1} \end{array} \\ \begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{array}$$

This matrix will be similar for $\beta_{(1)2}$. So, the $\kappa$ variables are set. For the $\theta$ variables, the steps are similar. To define the matrix for $\beta_{(2)1}$:

$$\beta_{(\mathbf{2})\mathbf{1}} = \begin{array}{c} \\ \eta_2 \\ \eta_1 \end{array} \begin{array}{c} \begin{array}{cccccc} \beta_{(1)0} & \beta_{(1)1} & \beta_{(1)2} & \beta_{(2)0} & \beta_{(2)1} & \beta_{(2)2} \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

which we simplify as:

$$\beta_{(\mathbf{2})\mathbf{1}} = \begin{array}{c} \\ \eta_2 \\ \eta_1 \end{array} \begin{array}{c} \beta_{(1)1} \\ \left[\begin{array}{c} 1 \\ 0 \end{array}\right. \end{array}$$

Like above, $\beta_{(2)2}$ will be the same, and this is a valid constraint matrix. For the intercepts, recall that the output is switched from what we would expect. So, the intercepts will be defined as:

$$\beta_{(\mathbf{2})\mathbf{0}}, \beta_{(\mathbf{1})\mathbf{0}} = \begin{array}{c} \\ \eta_2 \\ \eta_1 \end{array} \begin{array}{c} \beta_{(1)0} \quad \beta_{(2)0} \\ \left[\begin{array}{cc} 0 & 1 \\ 1 & 0 \end{array}\right. \end{array}$$

This matrix will constrain the intercepts properly as well as the other variables. It is crucial to research how the algorithm utilizes the constraints; otherwise, there could be errors and a failure to create an appropriate model. So, the constraint matrices for the variables are defined as:

$$\beta_{(\mathbf{2})\mathbf{0}}, \beta_{(\mathbf{1})\mathbf{0}} = \begin{array}{c} \\ \eta_2 \\ \eta_1 \end{array} \begin{array}{c} \beta_{(1)0} \quad \beta_{(2)0} \\ \left[\begin{array}{cc} 0 & 1 \\ 1 & 0 \end{array}\right. \end{array}$$

$$\beta_{(\mathbf{1})\mathbf{1}} = \begin{array}{c} \\ \eta_2 \\ \eta_1 \end{array} \begin{array}{c} \beta_{(1)1} \\ \left[\begin{array}{c} 0 \\ 1 \end{array}\right. \end{array}$$

$$\beta_{(\mathbf{1})\mathbf{2}} = \begin{matrix} \\ \eta_2 \\ \eta_1 \end{matrix} \begin{matrix} \beta_{(1)1} \\ \begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{matrix}$$

$$\beta_{(\mathbf{2})\mathbf{1}} = \begin{matrix} \\ \eta_2 \\ \eta_1 \end{matrix} \begin{matrix} \beta_{(1)1} \\ \begin{bmatrix} 1 \\ 0 \end{bmatrix} \end{matrix}$$

$$\beta_{(\mathbf{2})\mathbf{2}} = \begin{matrix} \\ \eta_2 \\ \eta_1 \end{matrix} \begin{matrix} \beta_{(1)1} \\ \begin{bmatrix} 1 \\ 0 \end{bmatrix} \end{matrix}$$

When we constrain the variables and make the model, the output is:

| Variable | Kappa | Theta |
|---|---|---|
| Kappa Intercept | 0.817 | 0.000 |
| Theta Intercept | 0.000 | 1.052 |
| | | |
| Witch Level | 0.000 | -0.046 |
| Soldier Level | 0.000 | -0.070 |
| Number of Witches | -0.015 | 0.000 |
| Number of Soldiers | -0.032 | 0.000 |

The VGLM model gave estimates that were *incredibly* close to the actual values. Because the VGLM can model two-parameter distributions, it is possible to specify which variables belong to which parameter using constraint matrices. The constraint matrices make the VLGM incredibly useful because we get thorough estimates. Although it is possible to use a transformed CLM

model, there is no way to specify a two-parameter distribution and get results for the corresponding independent variables. Furthermore, although using the GLM is easier with exponentially-distributed data, it is still a one-parameter estimator and also gives an additive result. Therefore, when dealing with more than one parameter, it is clear the choice is to use a VGLM.

# 4 An Example Using Real Data

The specialties of the VLGM model are now more clear; it is possible to model data outside of the exponential family, and the VGLM can model more than one linear predictor using constraint matrices. Let us look at a real data example and apply the VGLM. These data were collected by Professor Solomon at the State University of New York at Albany.

The data consist of records of the number of words read correctly by individual students over time. Variables included the student identifier, the timing of the probe (test), the number of words read correctly per minute by the student, and the teacher and grade of the student. In this case, the dependent variable will be words read correctly per minute. We can determine this follows a gamma distribution because there is a measurement over time of the words read.

The independent variable will be the time of the test (probe timing). Rather than using `GammaR`, the family used will be `Gamma2`, which will provide estimates of the expected value and shape parameter. The constraint

matrix is defined as:

$$\mathbf{H} = \begin{array}{c} \\ \eta_1 \\ \eta_2 \end{array} \begin{array}{c} \begin{array}{ccc} \beta_{(1)0} & \beta_{(1)1} & \beta_{(2)0} \end{array} \\ \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array}$$

The constraints for the variables can be specified:

$$\beta_{(\mathbf{1})\mathbf{0}}, \beta_{(\mathbf{2})\mathbf{0}} = \begin{array}{c} \\ \eta_1 \\ \eta_2 \end{array} \begin{array}{c} \begin{array}{cc} \beta_{(1)0} & \beta_{(2)0} \end{array} \\ \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \end{array}$$

$$\beta_{(\mathbf{1})\mathbf{1}} = \begin{array}{c} \\ \eta_1 \\ \eta_0 \end{array} \begin{array}{c} \begin{array}{c} \beta_{(1)1} \end{array} \\ \begin{bmatrix} 1 \\ 0 \end{bmatrix} \end{array}$$

The resulting model is:

| Variable | Kappa | Theta |
|---|---|---|
| Kappa Intercept | 4.297 | 0.000 |
| Theta Intercept | 0.000 | 1.040 |
| Probe Timing | 0.003 | 0.000 |

More interesting than the output of the model, which is difficult to directly interpret, is the graphic (Figure 10). The model can be specified for each student measured in the test. Then, we can evaluate the performance visually on the graphic. The red points represent Student 1; and the green, Student 2. The line corresponding to the colors of the student represents their progress estimated by the VGLM model. The black line is the model of probe timing for

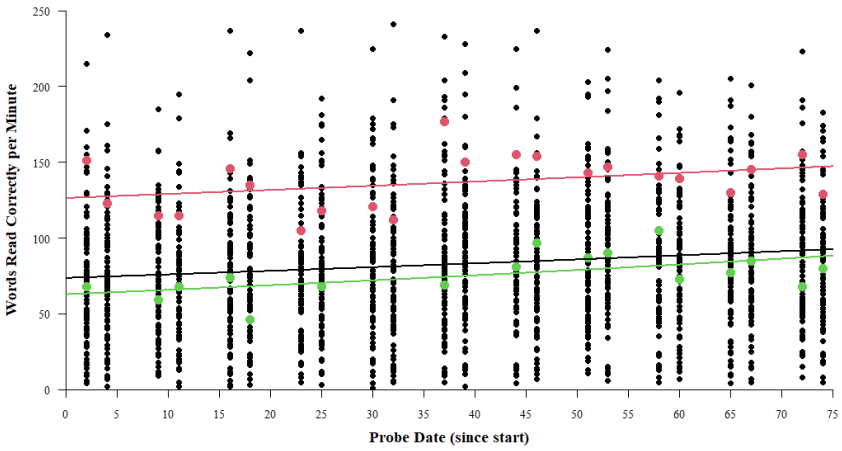


**Figure 10:** *Graphic of the results of the VGLM Real Data Model. The red dots and curve correspond to estimates for Student #1, while the green dots and curve correspond to estimates for Student #2. The black curve represents the average for all students in the sample.*

the overall data. Notice how well the lines fit the students' data; a graphic this nice is something that would not be possible with the GLM or CLM because there is more than one linear predictor in place here. Likely, the lines would not represent the students' progress as well as that of the VGLM.

The VGLM is very powerful for real-life data because in most cases, there is more than one linear predictor in place. The constraint matrices are an incredibly versatile component to specify which variables are for which parameters and get the proper estimates for the parameters of interest. Having more than one linear predictor, as well as constraint matrices, are some of the most important qualities that set the VGLM apart from the CLM and the GLM.

# 5 Conclusion

The goal of this project was to study the similarities and differences of the CLM using OLS, the GLM, and the VGLM. Furthermore, another goal was to understand not only how or why we use these models, but when we should use them as well as when the situation arises to use them. The literature review showed the requirements of the CLM, the GLM, and the VGLM. Many factors go into creating the CLM, and in most cases, it is challenging to find transformations that allow a statistician to get reasonable conclusions from the data. The GLM model lets the researcher model distributions that violate CLM requirements and also give the researcher an easier way to model dependent variables that follow distributions in the exponential family. It is also computationally inexpensive rather than having to do a plethora of transformations for the CLM.

The VGLM extends the GLM so the researcher can model data outside of the exponential family, which we saw worked nicely in the case of overdispersion. The VGLM can be computationally expensive because the IRLS algorithm takes longer to fit the model, but it is well worth it to have reasonable and more accurate estimations. Another valuable benefit of the VGLM model is its ability to model more than one linear predictor. In a lot of real-life applications, such as testing at schools, medical research, and economics, there is more than one linear predictor. The CLM model and the GLM model cannot fit this type of data efficiently; but, using constraint matrices, the VGLM can specify which variables belong to which parameters to get an accurate model and estimations.

When the data meet the requirements for the CLM, it is the best choice since it is easy to explain and interpret, is computationally efficient, and almost all statistical programs have a way to use this model. If the distribution of the

| Model | Pros | Cons |
|---|---|---|
| CLM | Easy to understand and reliable. | In real-life data, it is hard to meet all requirements and find an appropriate transform. |
| GLM | Fast and flexible with exponential family. | Need prior information of distributions and link functions. |
| VGLM | Flexible, not limited to a class of distributions, and more than one linear predictor. | Can be slow; as a relatively new model, it is also harder to understand. |

**Table 1:** *Pros and cons of each model.*

data happens to be in the exponential family, then the GLM is a good choice because it is a lot faster and more straightforward than using transformations in the CLM. Not all statistical programs have GLMs, but most of them do. When it is not possible to use a GLM or OLS, for example, in the case requirements are violated for both types of modeling or there is information to be known on more than one linear predictor, then using the VGLM becomes a wise choice. It is a bit more complex and is limited to fewer statistical applications, but is incredibly versatile and can help model far more than OLS and the GLM.

There are many pieces to the statistical modeling puzzle; there are more ways to create accurate models than one would ever believe is possible. The three I have used are the CLM, the GLM, and the VGLM, which was the reason for doing this project; I have seen each of these ways but lacked an understanding of why and when I would want to use them. After practicing modeling with the three models and doing a literature review, I am more practiced and well-versed in the ways of statistical modeling.

# References


[BP79] Trevor S. Breusch and Adrian R. Pagan. A simple test for heteroskedasticity and random coefficient variation. *Econometrica*, 47:1287–1294, 1979.

[Bra68] James V. Bradley. *Distribution-Free Statistical Tests*. Prentice Hall, 1968.

[For20] Ole J. Forsberg. *Linear Models and Řuritá Království : Using the Kingdom for Greater Insight*. https://rur.kvasaheim.com/, 0.704442d edition, 2020.

[LL22] Yang Liu and Baoding Liu. Estimating unknown parameters in uncertain differential equation by maximum likelihood estimation. *Soft Computing*, 26:2773–2780, 2022.

[NW72] John A. Nelder and R.W.M. Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society, Series A*, 135:370–384, 1972.

[SW65] Samuel Sanford Shapiro and Martin B. Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52:591–611, 1965.

[Woo06] Simon N. Wood. *Generalized Additive Models: An Introduction with R*. Texts in Statistical Science. Chapman and Hall/CRC, 2006.

[Yee15] Thomas W. Yee. *Vector Generalized Linear and Additive Models: With an Implementation in R*. Series in Statistics. Springer, 2015.