ST662 - Advanced Regression Analysis - Final Project Exploration and Application of Binomial GLM's

Douglas Bowen (160303920)

Date: 29-Mar-2022

Abstract

The objective of this report is to first and foremost discuss the theory behind the Binomial GLM and choices behind the link functions available. The next focus of this report is to apply this theory & binomial GLM links to a real-life dataset for predicting diabetes.

Two model methods are applied in this report. The first is the GLM as mentioned above, while the second is a similar GAM version. Each of these models will be analyzed, have coefficients tested, and have models compared. Eventually a final model will be selected.

The final result of this analysis found that the GLM and GAM models utilizing Complementary Log Log Link functions yielded near identical results with GAM being slightly better due to smoothing. The best predictors for a diabetes diagnosis ended up being Cholesterol, Weight, Age (Smoothed), and HDL.

Introduction

Overview

This project will examine a mix of theory and application of said theory - first reviewing some basics already taught to us, then expanding on this to topics that were left undiscussed. After theory is examined, an example will be given in applying said theory to a real-world dataset.

The first and primary objective of this report is to discuss the theory behind the Binomial GLM (and subsequently GAM) - specifically, how and why we derive/utilize the binomial GLM, as well as how to derive, differentiate, and choose an appropriate link function. Understanding the why and how for GLM's is of utmost important so that one can aptly determine the best model for their data in advance, and also avoid making any vital blunders or mistakes in model selection.

The second objective of this report is to apply said theory to a dataset comprised of over 400 individuals, with various metrics that are expected to give insight into a diabetes diagnosis or not. Medical diagnosis is an incredibly important ability to have, and while diabetes can be diagnosed directly through a blood test, being able to predict it off other easily-measurable factors (or at least get a good sense of risk level), is vital in preventing the risk of diabetes in advance. For example if someone knows that age and weight impact the odds of diabetes, they could periodically check their age/weight and see what likelihood that estimates them of having diabetes. This way, they'd know when they're statistically at risk and can make changes in advance before diabetic issues occur.

The dataset is discussed in detail in the Data Analysis section. As a quick overview, the dataset contains 19 variables. Almost all of them pertain to body measurements or statistics, while one is simply the location of the individual while another is an unidentifiable ID. The majority of the data is discrete integers, with a few continuous/factor variables. These variables will be vital in predicting diabetes, as evidenced in other real-world studies done.

Statistical Analysis

The statistical analysis performed combined a few tests together to get a finalized model. Generally speaking, the first step will be to determine an appropriate distribution family and accompanying link function for the data. Once this is done, a maximal (full) model will be constructed and checked for adequacy. Next, a nested (reduced) model will be selected after some testing on coefficients and compared with the original full one. This may be performed multiple times. Eventually, a final adequate model will be selected. The same tests will be performed for a GAM model, resulting in a final GLM and GAM model to compare. These will be compared while keeping in mind the "cost" of predictors in yielding their results so as to select the one that gives best results per predictor, so to speak.

Project Format

Please see the table of contents on the next page.

Table of Contents (TOC)

[1] Please note these sections on the report outline were meant to be separate, but that I've condensed them into these different portions with Professor approval.

Discussion of Theory

Background Refresher

Simple/Multi Linear Regression

In Linear Regression, we might take the following formula:

$Y = \beta_0 + \beta_1 x_1 + ... + \beta_p x_p + \epsilon$ with $\epsilon \approx N(0, \sigma_\epsilon^2)$

If we take the expected value of Y conditional on X, we could also write this as:

$E[Y|X] = \beta^T X$ (if we assume $x_0 = 1$)

In this formula, we have unknown constants $\beta_i, \sigma_\epsilon^2$

The error term ϵ varies randomly, but no matter what value our X takes on, the variance of our error does not change.

For linear regression, some of the major assumptions made are that:

1) Linearity - X and Y have a linear relationship.
2) Homoscedasticity - Variance of error term is the same for any X.
3) Normality of Errors - Error term is normally distributed.

However, when these assumptions do not hold true, Linear Regression can produce inaccurate or even nonsensical results. Hence, Generalized Linear Models are introduced (mainly in the case of (2)/(3), while Generalized Additive Models are utilized when (1) does not hold true).

Generalized Linear Models

We now construct regression models for when the assumptions mentioned above might not hold true. It takes the following formula:

$\eta = g(.) = g(\mu) = g(E[Y|X]) = \beta^T X$

We have our $X \in \mathbb{R}$ and some Y. This formula has three key components:

1) Random Component - Distribution for Y conditional on X, or $Y|X$
2) Systematic Component - Relates some parameter η to our X's
3) Link Component - Connects the random and systematic components together ($g(u) = \eta$)

With this in mind, our link component (function) is what allows us to map a non-linear relation to a linear one.

Linear Model: In the case of our aforementioned linear model:

1) Random Component - $Y|X \approx N(\mu, \sigma^2)$
2) Systematic Component - $\eta = \beta^T X$
3) Link Component - $g(u) = \mu$

Thus we see $\eta = g(\mu) = \beta^T X$ becomes $\mu = \beta^T X$

Or in other words, the expected value of Y conditional on X is equal to our systematic component.

Acceptable Density Functions

One important assumption for our generalized linear models (GLM's, henceforth) is that our random component $(Y|X)$ is assumed to have a probability density/mass function (PDF/PMF, henceforth) of the form:

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

where θ, ϕ are parameters and a, b, c are functions of said parameters.

Probability density functions of this form are home to the *exponential family* and called *exponential dispersion models*.

It should be noted that the expectation of said distribution is $E[Y|X]$.

Additionally:

- $E[Y|X] = \mu = b'(\theta)$
- $Var[Y|X] = b''(\theta)a(\phi)$

Background Theory Wrap-Up

For simple linear regression, we assume Homoscedasticity & Normality of our error terms.

When these assumptions change, we utilize a more generalized model, the generalized linear model (GLM).

$$\eta = g(E[Y|X]) = g(\mu) = \beta^T X$$

To summarize some notational aspects, we have:

- $E[Y|X]$ is what we want to estimate.
- θ determines the shape of our density for $Y|X$.
- ϕ is our dispersion parameter for the density of $Y|X$.
- $g(.)$ is our link function that maps a non-linear relationship to a linear one.
- η is our parameter used to model our transformation of $E[Y|X]$ by a function X (for GLM this is simply $\eta = \beta^T X$, a linear transformation).

Acronyms From here out, the following acronyms/short-hands will be utilized frequently.

- GLM = Generalized Linear Model
- E[Y|X] is simplified to E[Y] (conditionality is expected)
- GAM = Generalized Additive Model
- PDF/PMF = Probability Density/Mass Function
- EDM = Exponential Dispersion Model

Binomial GLMs

There are two cases in which our variance is not constant with respect to the binomial GLM. In both cases, $Y \in [0, 1]$.

(1) In the first case, Y might be a proportion (such as a total number of counts).

(2) In the second case, Y might be a binary value (0 for false, 1 for true).

As our Y is a value bounded by $[0, 1]$, our variances near said boundaries must automatically be smaller than those near the middle of the range. Thus, our variance is not constant (Homoscedasticity assumption fails). Furthermore, since we're bounded by $[0, 1]$, it would not be possible for our error (randomness) to be normally distributed (normality of error terms assumption fails).

And so we **shouldn't** use a simple linear regression model and instead must utilize GLM's.

So, we must now map some $Y \in \mathbb{R}$ to $Y \in [0, 1]$.

If we were inclined to, we could technically utilize a simple linear model - however, it would likely provide very poor results. For example:

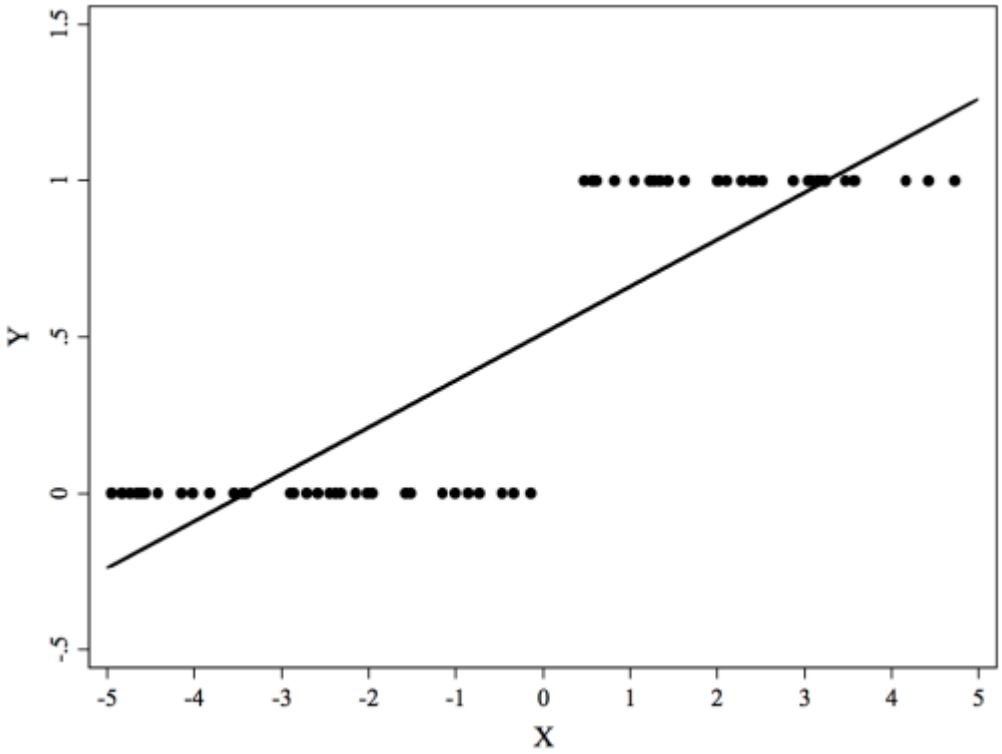


Figure 1: Source: Professor Ryan Bakker, POLS8501, UGeorgia

Support for Binomial GLM over Simple Linear Regression

To prove the above assumptions mathematically and provide support against utilizing a simple linear model, we examine theoretical binary data utilizing said simple linear regression model:

$$E[Y] = \beta^T X$$

In the case of binary data, we find our expected value of Y to be:

$$E[Y] = P(Y = 1)(1) + P(Y = 0)(0)$$
$$E[Y] = P(Y = 1)$$

And so our simple linear regression model can be represented as:

$$P(Y = 1) = \beta^T X$$

Which is simple to interpret.

However, we find that our variance is:

$$Var[Y] = E[Y](1 - E[Y])$$
$$Var[Y] = \beta^T X(1 - \beta^T X)$$

And thus our variance for Y is clearly not homoscedastic as it depends on X and is not constant.

Similarly, we can examine our error term ϵ. We know that our actual value for Y will either be 0 or 1. And our predicted value $E[Y]$ would be $\beta^T X$.

Thus, our ϵ_i term would be:

- $1 - \beta^T X_i$ when the actual value is 1
- $0 - \beta^T X_i$ when the actual value is 0

And thus can clearly never be normally distributed.

Thus, we've now mathematically shown that our ideal assumptions for a simple linear model have failed.

To further support that a simple linear model would be poor beyond just our assumptions failing, as $Y \in \mathbb{R}$, it is possible to predict values for $Y \in (-\infty, 0) \cup (0, 1) \cup (1, \infty)$. Considering our data is binary 0,1, these predictions might not be too helpful.

Thus we assume our $Y|X$ follows a binomial distribution. But what about error terms?

Link Functions for Binomial Data

As discussed above, we want our $Y \in [0, 1]$ for binomial data (binary or similarly constructed proportion). And so, we must determine some link function to appropriately map Y.

The following link functions can all sufficiently map Y:

1) **The Logit Link:**
   $g(\mu) = \log(\frac{\mu}{1-\mu}) = \text{logit}(\mu) = \text{log-odds}(\mu)$
2) **The Probit Link:**
   $g(\mu) = \phi^{-1}(\mu) = \text{probit}(\mu)$
3) **The Complimentary Log-Log Link:**
   $g(\mu) = \log(-\log(1-\mu)) = \text{cloglog}(1-\mu)$

Each of the above link functions has a slight difference in when they are used, though for the most part the Logit link is used by default.

The Logit Link Function

The Logit Function is considered the canonical link function for Binomial data - that is to say, the default link function that is utilized. It is, generally speaking, well applicable to binomial data and would provide similar results to probit/cloglog in many cases - however, it is important to make the distinction of use cases even if this link is the default.

The Logit Link arises from the fact that our error terms will not be normally distributed nor constant across values of X. Our Y value may only have two possible values (0,1) and so similarly (again as shown above) the error terms can also only have two values for each X.

As a result of this revelation, we assume that the error terms are **Logistically Distributed**.

This link also allows for interpretation in terms of odds-ratio.

The Probit Link Function

When we take data and dichotomize it (make it binary), we inherently lose some of the information in performing this transformation. For example, the data could be of any form (discrete, continuous, categorical, etc.) but has now been converted to some proxy binary variable. Some proxy variable like this might be more aptly defined as a **latent variable** that is derived from some threshold.

For examples sake, let us consider a datapoint we will examine later on in this report. From the Diabetes dataset within the Faraway package, we find that **glycosolated hemoglobin greater than 7.0** is usually taken as a positive diagnosis of diabetes.

Thus we can create a latent variable for diabetes diagnosis, either the individual has diabetes or they don't. However, the underlying data for this proxy variable is a continuous number and could theoretically take on any value from 0 to 1 (in percentage form, or 0 to 100 non-percentage).

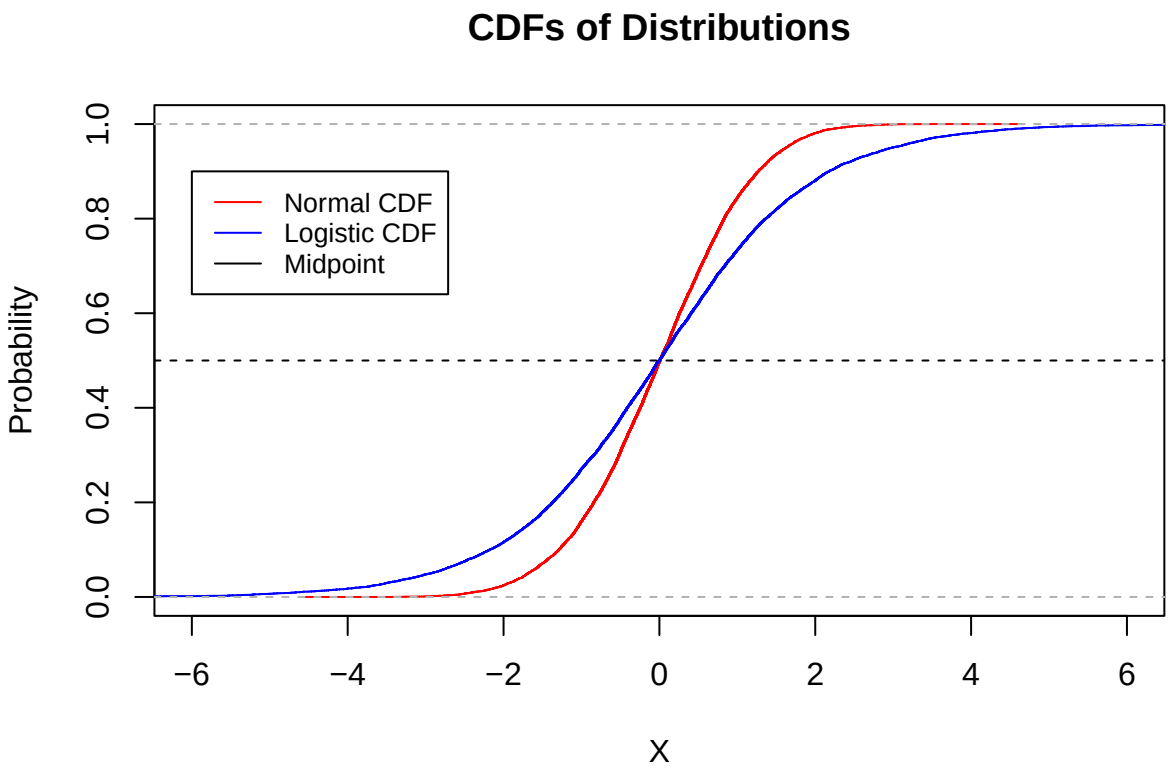
Lastly, if the true underlying variable that is being predicted is continuous, one can assume that the errors are in fact normally distributed and thus we can utilize a link function based off the normal CDF. As we utilize the inverse, this corresponds with a range between 0 and 1 inclusive.

Comparing the Logit and Probit Links

Both links are able to model a binary response with the only difference between them being the distributions of the error terms assumed for each.

It should be noted that when N is sufficiently large the binomial distribution sufficiently approximates the normal distribution.

Below we see a plot of the Normal and Logistic Distribution CDF's:



Clearly, we can see that both distribution CDF's are symmetrical around $Y = 0.5$. Furthermore, both lines are quite similar in shape/curve.

One final point to note is that the Logistic CDF is asymptotic as it approaches 0 and 1, while the normal distribution is well-defined.

This shines light on why the two links often produce incredibly similar results, except for when probabilities are near 0 or 1.

The Complimentary Log-Log Link Function

The third link function we examine is somewhat different from the prior two mentioned. The Complimentary Log-Log Link is much more rarely utilized and is specifically used for a niche type of data (generally speaking, survival analysis).

This third link selection is again based off of the assumed distribution of our error terms. Instead of assuming the error term follows a logistic or normal distribution, we assume that it follows the extreme-value distribution (e.g. Weibull).

As an illustrating example, take the injection of some chemical into an animal. We then examine the latent variable on if the animal survives the injection or not. In a case like this, very small changes in chemical dosage initially might not change the number killed much - but once the dosage reaches a certain point, this number killed jumps drastically. This is where the extreme-value distribution is handy.

The table below examines something like this - the number of beetles killed after injection of a certain dosage of a chemical. The different link functions predicted the following:

Log Dose	Number of Beetles	Number Killed	Fitted Values		
			Comp. Log-Log	Probit	Logit
1.691	59	6	5.7	3.4	35
1.724	60	13	11.3	10.7	9.8
1.755	62	18	20.9	23.4	22.4
1.784	56	28	30.3	33.8	33.9
1.811	63	52	47.7	49.6	50.0
1.837	59	53	54.2	53.4	53.3
1.861	62	61	61.1	59.7	59.2
1.884	60	60	59.9	59.2	58.8

Figure 2: Source: Professor K. Chough Carriere, STAT562, UAlberta

When we look at the lower doses, Probit/Logit links fail to produce good predictions (as compared to the complementary log log link). This is due to the impact of the proportion of beetles being killed jumping drastically between 1.784 and 1.811.

Threshold/Tolerance Distributions

A threshold (or tolerance) distribution is in essence what was discussed above - it is the distribution that you believe the latent variable may conform to, which in turn impacts the anticipated distribution that our error terms follow.

Now that the differences in the link functions has been explained, we can more formally define and derive said link functions. We will derive the probit link function utilizing the **Turbines** dataset that the textbook follows.

To preface, the information given to us on the Turbine dataset can be summarized here:

Hours (x)	Turbines (m)	Fissures (m*y)	Proportion (y)
400	39	0	0.0000000
1000	53	4	0.0754717
1400	33	2	0.0606061
1800	73	7	0.0958904
2200	30	5	0.1666667
2600	39	9	0.2307692
3000	42	9	0.2142857
3400	13	6	0.4615385
3800	34	22	0.6470588
4200	40	21	0.5250000
4600	36	21	0.5833333

Let us now assume that turbines have some tolerance (t_i) for hours used, and when this tolerance is below a certain threshold T, fissures develop. Assume this tolerance follows a normal distribution with mean tolerance τ_i. With this in mind, we have:

$$t_i \approx N(\tau_i, \sigma^2)$$
$$\tau_i = \beta_0' + \beta_1' x_i$$

We want to examine whether or not a turbine develops fissures, which we can write as:

$$y_i = \begin{cases} 1 \text{ if } t_i \leq T \text{ and we develop a fissure} \\ 0 \text{ if } t_i > T \text{ and we don't develop a fissure} \end{cases}$$

Thus, the probability of a fissure developing for a turbine can be written as:

$$\mu_i = E[y_i] = P(y_i = 1) = P(t_i \leq T) = \phi(\frac{T - \tau_i}{\sigma})$$

If we plug in τ_i, we find:

$$\frac{T - \tau_i}{\sigma} = \frac{T - \beta_0' - \beta_1' x_i}{\sigma} = \beta_0 + \beta_1 x_i$$

If we replace terms including β_0', β_1' as some $\beta_0 = \frac{T - \beta_0'}{\sigma}, \beta_1 = -\frac{\beta_1'}{\sigma}$, giving us our link function:

$$g(\mu_i) = \beta_0 + \beta_1 x_i$$

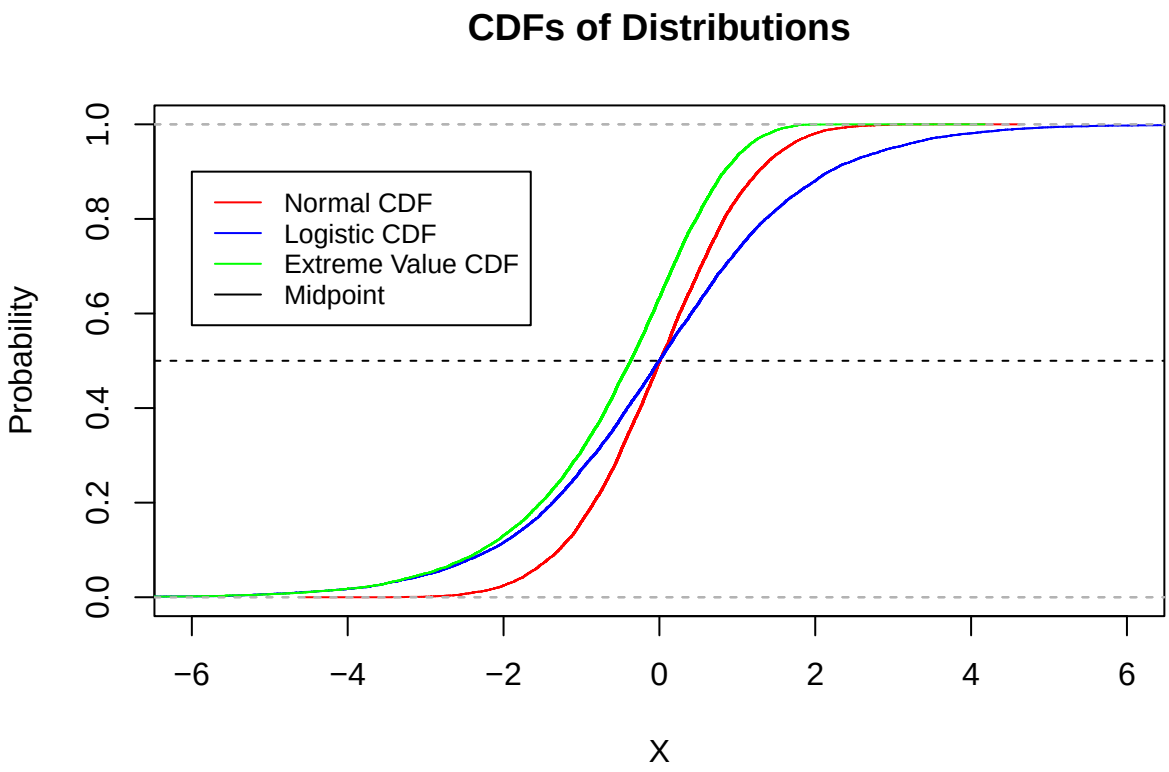
Similarly, one could derive the respective link functions for Logit or Complimentary Log-Log following this process.

Comparing the Common Link Functions

To wrap up this section pertaining to link function selection, we summarize the key differences below.

Link Type	Transformation	Threshold Distribution	Example Application
Logit	$\ln(\frac{\pi}{1-\pi})$	Logistic	Binary/Ordinal Data
Probit	$\phi^{-1}(\pi)$	Normal	Binary/Ordinal Data
C-Log-Log	$\ln(-\ln(1-\pi))$	Extreme Value	Survival Analysis

And we see the three CDF's plotted against each other:

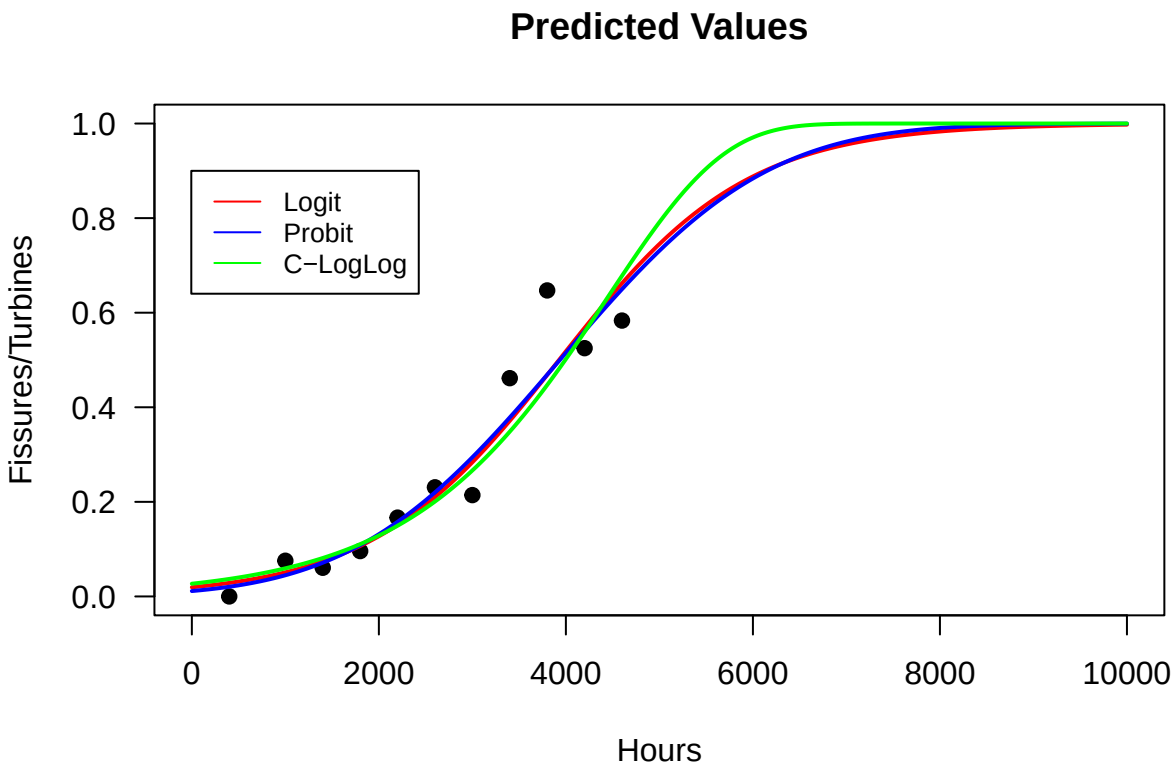


Again examining this plot, it becomes clear where slight differences in link functions exist.

Link Type	Curve Features	Characteristic	Potential Example
Logit	Symmetrical $Y \in (0,1)$	Equal Probability Change Rate P(Y=0) & P(Y=1) Do Not Exist	Undergrad vs. Grad
Probit	Symmetrical $Y \in [0,1]$	Equal Probability Change Rate P(Y=0) & P(Y=1) Do Exist	Turbine Fissures
C-Log-Log	Increasing $Y \in [0,1]$	Different Probability Change Rate P(Y=0) & P(Y=1) Do Exist	Dosage Survival

Illustrative Example

Before this section is wrapped up, we re-examine our turbine dataset against the different link functions. We see the results below, which all appear to give quite similar outcomes.



Key Take-Aways

As there has been plenty of theory discussed above, this section aims to quickly clarify the three key factors that must be specified in utilizing binomial GLM's.

1) The distribution of our response variable Y conditional on X (or, $Y|X$) is presumed to follow the binomial distribution.

2) The distribution of our error terms ϵ are presumed (commonly) follow one of three distributions - the logistic distribution, the normal distribution, and the extreme value distribution.

3) Based on the error term distribution, we construct our link function.

Generally speaking, the link functions will tend to return very similar results. However, certain niche cases can make one a better fit than others.

Data Analysis

Data Description

We will be utilizing the dataset **Diabetes** that is taken from the **Faraway** library.

Description: 403 African Americans were interviewed in a study to understand the prevalence of obesity, diabetes, and other cardiovascular risk factors in central Virginia.

Variable	Type	Description
ID	Discrete	Subject ID
Chol	Discrete	Total Cholesterol
Stab.Glu	Discrete	Stabilized Glucose
HD1	Discrete	High-Density Lipoprotein
Ratio	Continuous	Cholesterol/HDL Ratio
GlyHB	Continuous	Glycosolated Heomglobin
Location	Factor	County (Buckingham or Louisa)
Age	Discrete	Age in Years
Gender	Factor	Gender (Male or Female)
Height	Discrete	Height in Inches
Weight	Discrete	Weight in Pounds
Waist	Discrete	Waist in Inches
Hip	Discrete	Hip in Inches
Frame	Factor	Body Frame (Small/Medium/Large)
BP.1S	Discrete	First Systolic Blood Pressure
BP.1D	Discrete	First Diastolic Blood Pressure
BP.2S	Discrete	Second Systolic Blood Pressure
BP.2D	Discrete	Second Diastolic Blood Pressure
Time.PPN	Discrete	Post-Prandial Time when Labs were Drawn (Minutes)

It also denotes that "a glycosolated hemoglobin greater than 7.0 is usually taken as a positive diagnosis of diabetes".

Thus, we will use this as our latent variable to predict diabetes.

Data Cleaning/Adjustments

Cleaning:

The first step before building our models is cleaning our data. For starters, we will remove the ID column as this information is not useful to us.

Next, we examine any missing data to ensure we don't accidentally use a variable that has a majority of missing values.

A quick summary output was taken for our data which revealed first and foremost that 262/403 subjects did not get a second blood pressure reading. Instead of removing over half our subjects, will will instead remove these variables as they are not sufficiently populated.

We then re-examine our summary of our data and find that the most NA entries is 13 for GlyHB. Since this will be our latent variable that we're trying to predict based off of, we will remove these rows entirely. Our dataset is now of size 390.

This final adjustment now leaves us with 24 rows of data remaining that contain some form of NA value. The largest number of NA's now appears in the Frame variable (11 NA's), while the next highest number of NA values for a variable is just 5.

If our dataset were larger, leaving these NA values in still might be ok and make no discernable difference. However, as our dataset is quite small, we remove them and end up with 390 rows of useable data.

Dichotomizing Glycosolated Hemoglobin:

We now can dichotomize our latent variable Glycosolated Hemoglobin. We will replace the values in this column with a 1 or 0 as follows:

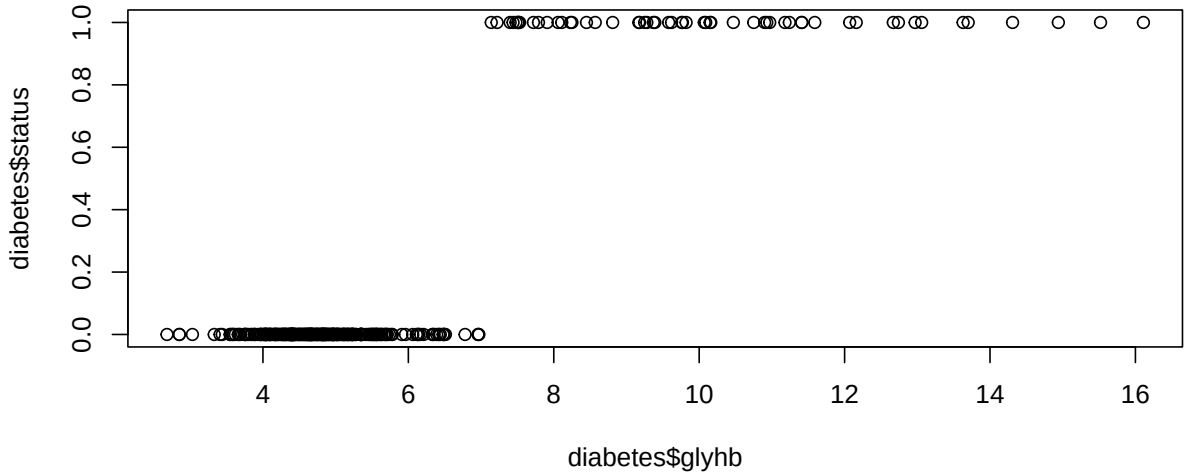
If GlyHB > 7, we designate the individual as having diabetes (1). Otherwise (≤ 7), they do not have diabetes (0).

We then rename the column "status" to represent diabetes status.

With our data now cleaned and adjusted, we can move on to examining the data more in-depth and determine some working models.

Model Building & Testing (GLM)

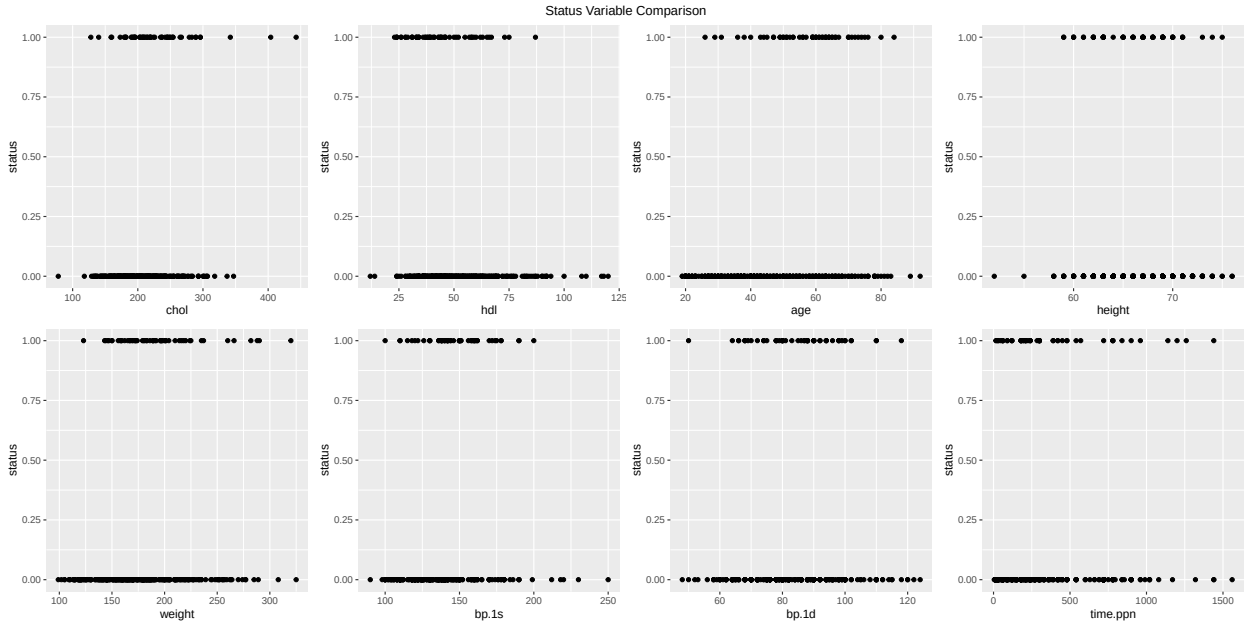
Before we begin building our models, we first examine our latent variable to see if we notice anything unique. From plotting our gly-hb levels against status classification, we see that there are more negative status results than positive ones by a large margin.



Though our latent variable was continuous which might suggest a probit link, our latent variable value range was not evenly split. Our minimum value was 2.6800001 while our maximum was 16.1100006. Yet, we've determined diabetes to be anything greater than 7. So non-diabetes values have a range of 4.3199999 while diabetic values have a range of 9.1100006, nearly double. This discrepancy might make our cloglog link more reasonable.

As a result, we will examine models with both the probit and cloglog links.

We next examine each variable (aside from factors) against the individuals status to check if anything pops out at us. We might want to see which plots lend themselves to an appropriate shape.



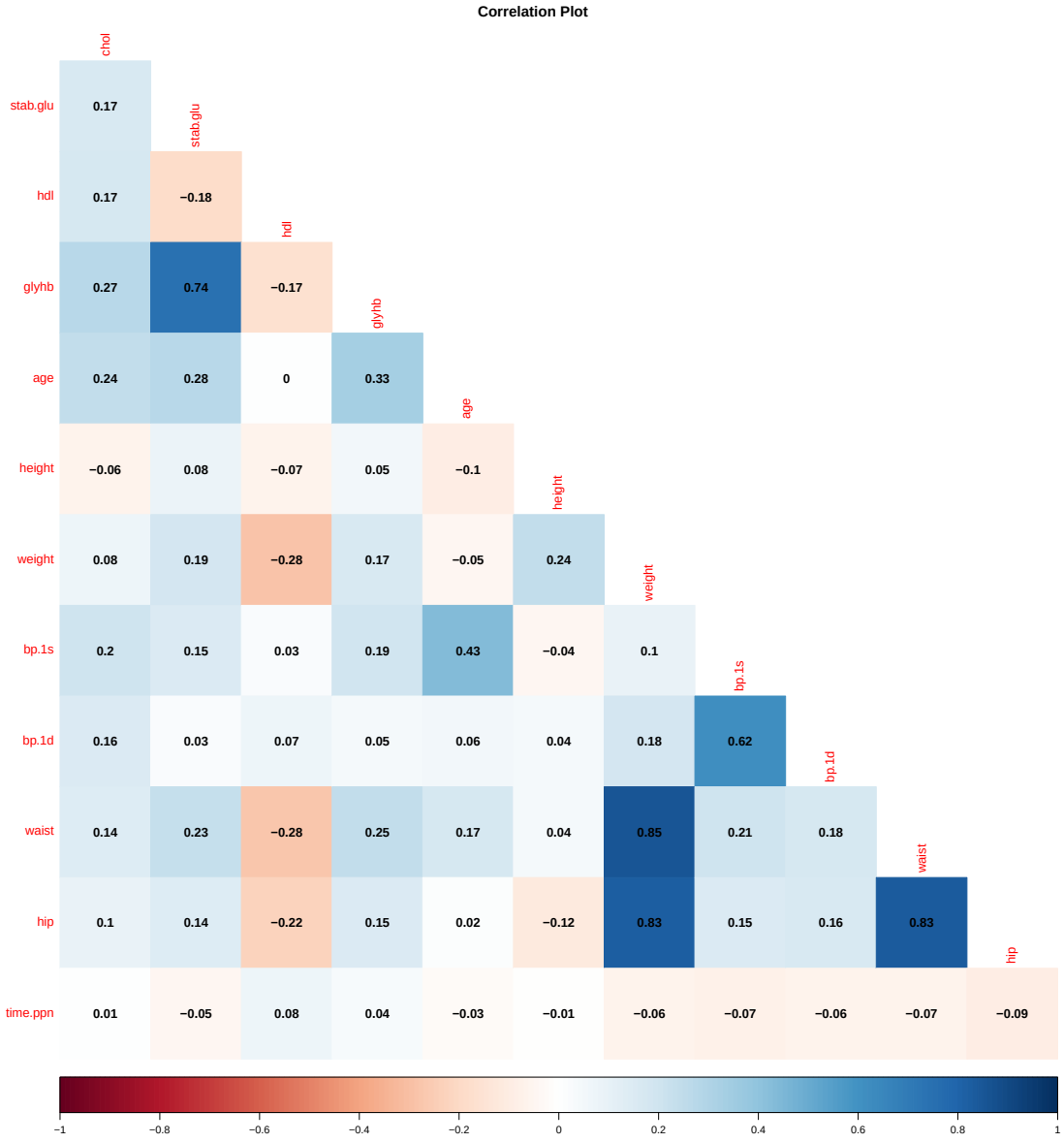
However, none appear to be very well-divided.

Correlation Check:

Furthermore, before constructing our models we want to examine if any variables are highly correlated as this might cause issues in said model (due to multicolinearity).

First and foremost, it should be noted that Ratio is a combination of Cholesterol / HDL, and so we remove this variable while keeping Cholesterol & HDL.

Moving on, a quick correlation plot reveals to us that {Waist, Hip, Weight} are all highly correlated to one another. Additionally, we also see that the variables {glyhb, stab glu} are highly correlated. Lastly, we find {bp.1d, bp.1s} to be moderately correlated.



With this in mind, we now create a very quick basic probit model with all predictors so we can quickly examine the VIF of each:

	GVIF	Df	GVIF^(1/(2*Df))
chol	1.134896	1	1.065315
stab.glu	1.111551	1	1.054301
hdl	1.241521	1	1.114236
location	1.207480	1	1.098854
age	1.703745	1	1.305276
gender	2.744714	1	1.656718
height	2.512097	1	1.584960
weight	6.831798	1	2.613771
frame	1.668854	2	1.136592
bp.1s	2.279666	1	1.509856
bp.1d	2.017329	1	1.420327
waist	4.997006	1	2.235398
hip	5.545282	1	2.354842
time.ppn	1.067179	1	1.033043

Clearly, we find {Waist, Hip, Weight} to be highly correlated as their GVIF values are much higher than 2. To account for this, we will remove the Hip & Waist predictors while leaving Weight, as generally speaking weight is known to be correlated to diabetes while waist/hip can simply be abnormally large without weight being high.

Though stabilized glucose does not have a high VIF, in researching it was found that stabilized glucose and glycosolated hemoglobin are highly correlated. As a result, we'll also remove stabilized glucose as it would be similar to the underlying variable that created our diabetes status.

Now knowing that we want to remove the aforementioned predictors, we will repeat our data-cleaning steps to see if we can slightly increase the data we have to work with (as there were a few NA rows exclusively in waist/hip/ratio). In doing so, we gain 2 rows of data to increase our total rows to 368.

Model Creation:

We now create a model to predict status using a binomial GLM. To start, we will simply examine the probit and cloglog models with all predictors aside from glyhb included (the maximal model) and then reduce from there.

We can see the summary of these two models here:

Probit Model:

```
Call:
glm(formula = status ~ (. - glyhb), family = binomial(link = "probit"),
    data = diabetes)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.4035  -0.5762  -0.3426  -0.1403   2.8956

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)    -6.3234988  2.7228289  -2.322  0.02021 *
chol            0.0061117  0.0020668   2.957  0.00311 **
hdl            -0.0191349  0.0063938  -2.993  0.00277 **
locationLouisa -0.0436512  0.1867154  -0.234  0.81515
age             0.0304404  0.0073298   4.153 3.28e-05 ***
genderfemale    0.1952993  0.2849166   0.685  0.49305
height          0.0237528  0.0365751   0.649  0.51606
weight          0.0064722  0.0027267   2.374  0.01761 *
framemedium    -0.2499391  0.2571474  -0.972  0.33107
framelarge     -0.2366073  0.3061160  -0.773  0.43956
bp.1s           0.0042520  0.0053630   0.793  0.42788
bp.1d          -0.0006483  0.0091226  -0.071  0.94335
time.ppn        0.0004285  0.0002953   1.451  0.14675
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 313.88  on 367  degrees of freedom
Residual deviance: 247.61  on 355  degrees of freedom
AIC: 273.61

Number of Fisher Scoring iterations: 6
```

Figure 3: Full GLM Probit Model Summary

CLogLog Model:

```
Call:
glm(formula = status ~ (. - glyhb), family = binomial(link = "cloglog"),
    data = diabetes)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.5309  -0.5358  -0.3401  -0.1906   2.7664

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)    -1.105e+01  4.446e+00  -2.486  0.01291 *
chol            9.205e-03  2.927e-03   3.145  0.00166 **
hdl            -3.323e-02  1.109e-02  -2.997  0.00273 **
locationLouisa -8.014e-02  2.845e-01  -0.282  0.77817
age             5.164e-02  1.149e-02   4.495 6.94e-06 ***
genderfemale    3.432e-01  4.536e-01   0.757  0.44929
height          4.743e-02  5.829e-02   0.814  0.41582
weight          1.026e-02  4.032e-03   2.545  0.01094 *
framemedium    -3.785e-01  4.252e-01  -0.890  0.37349
framelarge     -3.982e-01  4.817e-01  -0.827  0.40844
bp.1s           3.506e-03  7.945e-03   0.441  0.65901
bp.1d           5.418e-03  1.450e-02   0.374  0.70865
time.ppn        8.379e-04  4.420e-04   1.896  0.05803 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 313.88  on 367  degrees of freedom
Residual deviance: 246.63  on 355  degrees of freedom
AIC: 272.63

Number of Fisher Scoring iterations: 7
```

Figure 4: Full GLM CLogLog Model Summary

Based on these models, we find both provide the same significance conclusions. Based on the previously mentioned reasoning behind why the CLogLog link may be better, along with the fact that the p-values for CLogLog are slightly lower, we'll continue our analysis from here on out with just the CLogLog model.

Calculating our Coefficients (Beta):

As a very quick refresher, our coefficients are calculated using the iterative weighted least squares (IWLS) technique. In the above pictures, we can see the number of iterations required at the bottom. Though not specifically derived here, the general algorithm is given:

(1) Initialize the algorithm by selecting random starting values for $\beta^{(0)}$, the vector of all β's. A general starting point is often 0 or 1.

(2) Construct **adjusted** dependent variables $z^{(0)} = \Sigma_{k=1}^{p} x_{ik}\beta_k^{(0)} + (y_i - \mu_i^{(0)})(\frac{d\eta_i}{d\mu_i^{(0)}})$

(3) Construct our weights $w_{ii}^{(0)} = \frac{1}{Var(Y_i)}(\frac{d\mu_i}{d\eta_i})^2$

(4) Perform our weighted least squares calculation $\beta^{(1)} = (X^T W^{(0)} X)^{-1} X^T W^{(0)} Z^{(0)}$

(5) Repeat the steps until the change in iterated β's is approximately 0, that is $|\beta^m - \beta^{(m-1)}| = 0$.

Once this is done, you'll have estimates for our coefficients.

Coefficient Significance Test

In examining our model, we now find that Cholesterol, HDL, Age, and Weight are significant. That is to say we have the following hypothesis test performed:

$$H_0 : \beta = 0 \text{ vs. } H_A : \beta \neq 0$$

In which we reject the null hypothesis for the β relating to Cholesterol, HDL, Age, and Weight. For the other variables, we fail to reject the null hypothesis ($\beta = 0$). As a result of this test, we create the reduced model utilizing only the significant variables. The model is as follows:

```
Call:
glm(formula = status ~ chol + hdl + age + weight, family = binomial(link = "cloglog"),
    data = diabetes)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.4424  -0.5430  -0.3589  -0.1953   2.8032

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -6.974607   1.239612  -5.626 1.84e-08 ***
chol         0.009671   0.002749   3.519 0.000434 ***
hdl         -0.028277   0.010136  -2.790 0.005275 **
age          0.048145   0.009545   5.044 4.56e-07 ***
weight       0.010281   0.003514   2.926 0.003436 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 313.88  on 367  degrees of freedom
Residual deviance: 251.83  on 363  degrees of freedom
AIC: 261.83

Number of Fisher Scoring iterations: 6
```

Model Adequacy (Goodness of Fit): Maximal & Nested

Before we compare the two models to find which is the better use, we first check to ensure that they are both adequate using a goodness of fit test. The goodness of fit test is a statistical test that determines whether or not the observed values match those expected by the model.

For the goodness-of-fit test, we simply find our deviance and calculated chi-squared value. If this results in a p-value less than 0.05, we reject the model and say it is not adequate. If it is greater, we accept it as adequate.

For these two models, we find the following p-values:

(1) Full Model = 0.9999973

(2) Nested Model = 0.9999981

We find that both are adequate.

Model Selection (Likelihood Ratio Test): Maximal Model vs. Nested Model

We now want to examine if which model is better. It should be noted we do not use ANOVA because, as mentioned prior, residual deviance and pearson residuals are determined entirely by the fitted values and hence are not meaningful.

We utilize likelihood ratio tests (or score tests) for binary data. We will use the likelihood ratio test.

We calculate some deviance from our likelihood ratio in the form:

$$LRT = 2[\text{LogLik(Full Model) - LogLik(Nested Model)}]$$

We then perform the following hypothesis test:

$$H_0 : \beta_{q+1} = ... = \beta_p = 0 \text{ vs. } H_A : \text{At least one of } \beta_{q+1}...\beta_p \neq 0$$

Where β_{q+1} to β_p is the additional coefficients of the maximal model.

Our resulting p-value from said test is 0.7356438 which is of course greater than 0.05 and thus we fail to reject the null hypothesis. This means that we find our reduced model to be adequate and thus should utilize it over the maximal model.

Thus, our final model selected can be written as follows:

$$\log(-\log(1-\pi)) = \beta_{\text{intercept}} + \beta_{\text{age}}x_{\text{age}} + \beta_{\text{weight}}x_{\text{weight}} + \beta_{\text{hdl}}x_{\text{hdl}} + \beta_{\text{chol}}x_{\text{chol}}$$

Where the respective betas are $\{-6.974607, 0.048145, 0.010281, -0.028277, 0.009671\}$.

Model Building & Testing (GAM)

Coefficient Significance Test

Note: The Null Hypothesis and Alternative Hypothesis are the same as in the GLM section, with their respective coefficients. Hence it will not be re-displayed here.

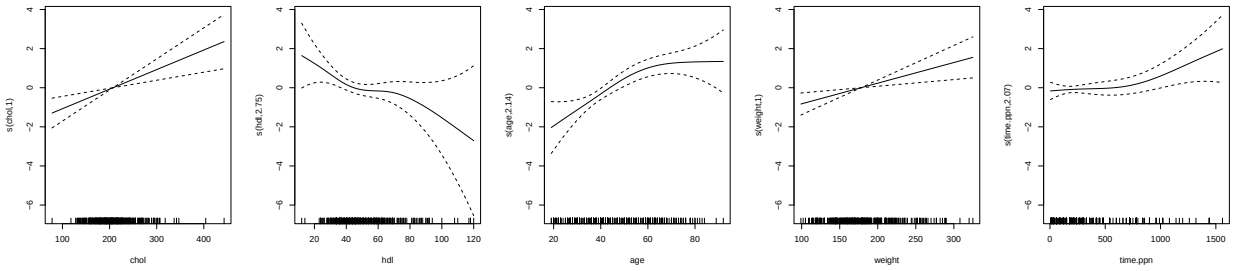
We will now examine a generalized additive model instead of a linear one. The first model we look at is again the maximal model, where we've applied a smoothing function to all non-categorical predictors (i.e. all but gender, frame, and location).

In examining the summary of our model, we find that the same predictors from GLM (Age, Weight, Cholesterol, HDL) are significant, along with adding time.ppn as well. Thus, we try reducing our model to just include these variables.

```
Family: binomial
Link function: cloglog

Formula:
status ~ s(chol) + s(hdl) + s(age) + s(height) + s(weight) +
    s(bp.1s) + s(bp.1d) + s(time.ppn) + location + gender + frame

Parametric coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)    -2.31478    0.54758  -4.227 2.36e-05 ***
locationLouisa -0.05771    0.29347  -0.197    0.844
genderfemale    0.37656    0.46435   0.811    0.417
framemedium    -0.46612    0.44137  -1.056    0.291
framelarge     -0.58658    0.50718  -1.157    0.247
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
             edf Ref.df Chi.sq p-value
s(chol)    1.000  1.000 10.108 0.00148 **
s(hdl)     3.664  4.614 12.040 0.02597 *
s(age)     2.018  2.565 15.286 0.00113 **
s(height)  1.000  1.000  0.140 0.70817
s(weight)  2.359  2.996  8.670 0.03327 *
s(bp.1s)   2.112  2.721  2.521 0.34759
s(bp.1d)   1.000  1.000  0.000 0.98483
s(time.ppn) 1.913  2.411  6.560 0.04954 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.249   Deviance explained = 28.4%
UBRE = -0.28003  Scale est. = 1         n = 368
```

Figure 5: Full GAM Model Summary

For the reduced model, we examine the plots of the smoothing functions to see if smoothing is necessary, or if the variable is significant sans smoothing:



From these plots, we notice that while HDL, Age, and Time.PPN clearly have some degree of non-linear form (i.e. curvature), Cholesterol and Weight are linear. This indicates they may be significant even without smoothing terms, and so we remove the smoothing to generate a third model.

Examining the summary of this model with no smoothing , we find that Time.PPN becomes non-significant. However, once this is removed, HDL also becomes non-significant (though quite close at 0.556). As a result, we try to see if HDL may still be significant when not smoothed. And thus we arrive at our final reduced model for GAM where Age is smoothed, while Cholesterol, HDL, and Weight are not.

```
Family: binomial
Link function: cloglog

Formula:
status ~ chol + hdl + s(age) + weight

Parametric coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.898566   1.089486  -4.496 6.92e-06 ***
chol         0.009281   0.002772   3.348 0.000814 ***
hdl         -0.025106   0.010167  -2.469 0.013537 *
weight       0.010308   0.003540   2.912 0.003595 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Approximate significance of smooth terms:
         edf Ref.df Chi.sq  p-value
s(age) 2.105  2.683  24.64 2.36e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

R-sq.(adj) =  0.194   Deviance explained = 21.7%
UBRE = -0.29876  Scale est. = 1         n = 368
```

Figure 6: Final GAM Model Summary

Model Adequacy (Goodness of Fit): Maximal & Nested

We again perform a goodness of fit test.

For these two models, we find the following p-values:

(1) Full Model = 1

(2) Final Model = 0.9999994

We find that both are adequate.

Model Selection (Likelihood Ratio Test): Maximal Model vs. Nested Model

We now confirm model adequacy again in the same fashion as done with the GLM's. We compare the "maximal" model and the "reduced" model where the reduced model in this instance is the final one (Age is smoothed, Cholesterol, HDL, and Weight are not).

Our resulting p-value from said test is 0.0996286 which is more than 0.05 and thus we fail to reject the null hypothesis. This means that we find our reduced model to be adequate and thus should utilize it over the maximal model.

Results

We now have two finalized models for the GLM method and the GAM method. As the models are separate and not nested in any way, one way to compare which is better is through AIC or BIC, in which a lower value provides a better model.

We find an {AIC, BIC} for each model of:

(1) GLM Model: {261.8323812, 281.3727959}

(2) GAM Model: {258.0574767, 281.9164893}

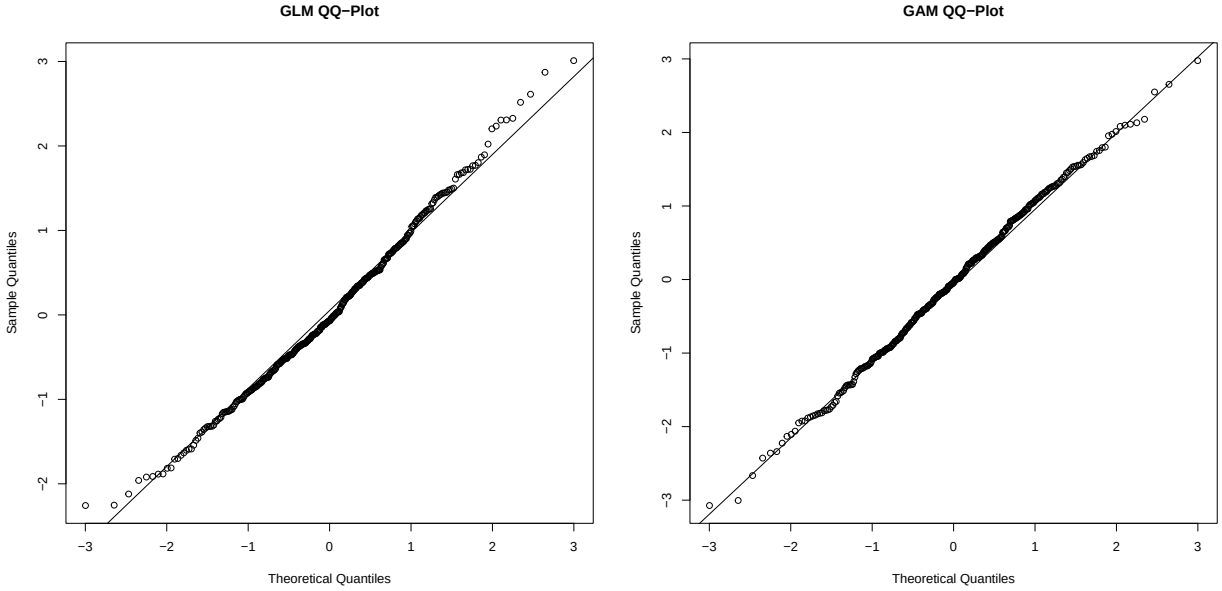
Here we clearly see that both models have quite close AIC/BIC scores - however, the AIC difference is larger than the BIC difference and favours the GAM.

As a quick explanation, AIC and BIC penalize models for excessive predictor usage, just in different ways.

(1) AIC = 2k - 2ln(Maximum Likelihood) where k is the number of predictors/parameters.

(2) BIC = -2ln(Maximum Likelihood) + k*ln(n) where n is the number of observations/rows.

We also examine a QQ-Plot (which provides an indication of normality) for these models:



Though quite similar, this also shows us that GAM is an ever so slightly better fit.

Realistically, both models appear to be nearly identical in comparison of AIC/BIC and QQ-Plot, however, we will choose the GAM due to the slight difference.

Both finalized/reduced models found the same predictors to be significant. The only difference in the end was applying a smoothing function to age.

Conclusion & Discussion

Strengths

For starters, this analysis utilizing GLM's and GAM's ended up concluding that the same predictors were significant, which is, generally speaking, a good confirmation check (especially since the full GAM only had non-significant values that were linear as opposed to smoothed).

The analysis also concluded that the nested model was better in all cases which is what one would hope to see when removing non-significant predictors.

Most importantly, the predictors chosen as indicators for diabetes were (upon external research) ones that have already been studied and found to have links to diabetes, which indicates that we've likely chosen a very strong model. Furthermore, most predictors that were **not** selected did not have studies examining links, or had studies that found no link as well!

Shortcomings

The major shortcoming of this analysis is simply in the underlying knowledge required to understand the dataset. Even with external research, my knowledge of the predictors and what they actually meant medically or in the context of diabetes was quite poor, and, as a result, there may have been mis-steps in handling the data or performing the analysis. As an example, without external research I would not have found that stabilized glucose would be a very correlated value to glycosylated hemoglobin and would have accidentally been included in the model (which would cause quite a lot of issues).

Another shortcoming, albeit less major, was the size of the dataset. Having only 368 subjects with data is not great in general. When trying to predict medical accuracy, it is important to be precise - with less data, we introduce a higher risk of incorrect conclusions being drawn due to such a small subset of data. For example we may find with 3,000 samples that some other predictor is quite significant but which we missed due to such a small sample. Additionally, we removed important data due to high missingness.

Lastly, the analysis took liberties and made assumptions on link function instead of examining all of them - as a result, another link may have ended up providing a better model but was missed due to the selection process.

References/Works Cited

(1) Generalized Linear Models, (Tibs, Ryan).
Link: https://www.stat.cmu.edu/~ryantibs/advmethods/notes/glm.pdf

(2) Complementary Log-Log Model, (Carrier, K)
Link: http://www.stat.ualberta.ca/~kcarrier/STAT562/comp_log_log.pdf

(3) Binary, Logit, and Probit Links, (Bakker, R)
Link: https://spia.uga.edu/faculty_pages/rbakker/pols8501/MLENotes2a.pdf

(4) Generalized Linear Models, Link Functions, (Newsom, J)
Link: http://web.pdx.edu/~newsomj/cdaclass/ho_glm.pdf

(5) Link Functions and the Generalized Linear Model, (Newsom, J)
Link: http://web.pdx.edu/~newsomj/mvclass/ho_link.pdf

(6) Generalized Linear Models, Other Choices of Link (Rodriguez, German)
Link: https://data.princeton.edu/wws509/notes/c3s7

(7) Dataset: Faraway Library, "Diabetes"

(8) Textbook: Generalized Linear Models with Examples in R, {(Dunn, Peter), (Smyth, Gordon)}

Appendix

Code

See R Markdown File if you would like to examine code in-depth.

```
# Setup Chunk
knitr::opts_chunk$set(echo = TRUE, tidy.opts = list(width.cutoff = 80), tidy = TRUE)

# Required Libraries for Course
library(GLMsData)
library(aod)  #For Quick Wald Test
library(lmtest)  #For Quick Log Likelihood
library(statmod)  #For Quantile Residual
library(KernSmooth)  #A3
library(splines)
library(lmvar)
library(ISLR)
library(corrplot)
# library(secr)

# Optional Libraries Not Recommended By Course Libraries for Graphing
library(tidyverse)
library(scales)
library(gridExtra)

# Other Useful Libraries (for R Markdown)
library(knitr)
library(kableExtra)
library(latex2exp)
library(tinytex)

# Old+Useful Packages from Prior Courses library(multcomp)
# library(multcompView)
library(pander)
library(MASS)

## Multiple Fractional Polynomial Model
library(mfp)
## Generalized Additive Models
library(mgcv)
## Multivariate Adaptive Regression Spline (MARS) Models
library(earth)


# install.packages('EnvStats')
library(EnvStats)

library(faraway)
# data('diabetes', package='faraway') ?diabetes
library(GGally)
library(regclass)

library(GLMsData)
```

```r
data(turbines, package = "GLMsData")

# Kable R Summary Output
library(sjPlot)
library(sjmisc)
library(sjlabelled)


# Notes: Use 'install.packages('library_name_here')' then use
# 'library(library_name_here)'
normal_dat = ecdf(rnorm(10000))
logi_dat = ecdf(rlogis(10000))
extrm_dat = ecdf(-revd(10000))

plot(0, 0, pch = "", ylim = c(0, 1), xlim = c(-6, 6), xlab = "X", ylab = "Probability",
    main = "CDFs of Distributions")
lines(normal_dat, col = "red")
lines(logi_dat, col = "blue")
abline(h = 0.5, col = "black", lty = 2)
legend(-6, 0.9, legend = c("Normal CDF", "Logistic CDF", "Midpoint"), col = c("red",
    "blue", "black"), lty = 1, cex = 0.8)
turbines$Proportion <- turbines$Fissures/turbines$Turbines
colnames(turbines) <- c("Hours (x)", "Turbines (m)", "Fissures (m*y)", "Proportion (y)")
plot(0, 0, pch = "", ylim = c(0, 1), xlim = c(-6, 6), xlab = "X", ylab = "Probability",
    main = "CDFs of Distributions")
lines(normal_dat, col = "red")
lines(logi_dat, col = "blue")
lines(extrm_dat, col = "green")
abline(h = 0.5, col = "black", lty = 2)
legend(-6, 0.9, legend = c("Normal CDF", "Logistic CDF", "Extreme Value CDF", "Midpoint"),
    col = c("red", "blue", "green", "black"), lty = 1, cex = 0.8)

data(turbines, package = "GLMsData")

glm.fit.1 <- glm(Fissures/Turbines ~ Hours, family = binomial(link = "logit"), weight = Turbines,
    data = turbines)
glm.fit.2 <- glm(Fissures/Turbines ~ Hours, family = binomial(link = "probit"), weight = Turbines,
    data = turbines)
glm.fit.3 <- glm(Fissures/Turbines ~ Hours, family = binomial(link = "cloglog"),
    weight = Turbines, data = turbines)

newHour <- seq(0, 10000, length = 1000)
newMab1 <- predict(glm.fit.1, se.fit = TRUE, newdata = data.frame(Hours = newHour))
newMab2 <- predict(glm.fit.2, se.fit = TRUE, newdata = data.frame(Hours = newHour))
newMab3 <- predict(glm.fit.3, se.fit = TRUE, newdata = data.frame(Hours = newHour))

plot(Fissures/Turbines ~ Hours, data = turbines, xlim = c(0, 10000), ylim = c(0,
    1), las = 1, pch = 19, main = "Predicted Values")
lines(exp(newMab1$fit)/(1 + exp(newMab1$fit)) ~ newHour, lwd = 2, col = "red")
lines(pnorm(newMab2$fit) ~ newHour, lwd = 2, col = "blue")
lines(1 - exp(-exp(newMab3$fit)) ~ newHour, lwd = 2, col = "green")
legend(-6, 0.9, legend = c("Logit", "Probit", "C-LogLog"), col = c("red", "blue",
    "green"), lty = 1, cex = 0.8)
```

```r
data("diabetes", package = "faraway")
diabetes <- diabetes


diabetes <- subset(diabetes, select = -c(id))

# Initial
summary(diabetes)
diabetes[!complete.cases(diabetes), ]

# Remove 2nd Reading
diabetes <- subset(diabetes, select = -c(bp.2s, bp.2d))

# After removing second BP readings
summary(diabetes)
# Remove remaining NA rows
diabetes <- diabetes[complete.cases(diabetes), ]
summary(diabetes)

# Dichotomize GlyHB, Rename
diabetes$status <- diabetes$glyhb
diabetes[which(diabetes[, "glyhb"] <= 7), "status"] = 0
diabetes[which(diabetes[, "glyhb"] > 7), "status"] = 1


# Convert to Factor diabetes$status <- as.factor(diabetes$status)

# Final Summary
summary(diabetes)

plot(diabetes$glyhb, diabetes$status)
# ggpairs(diabetes)
gg_male <- ggplot(data = diabetes, aes(y = status))
# gg_male <- ggplot(data=diabetes[which(diabetes$gender=='male'),],
# aes(y=status))

Status1 <- gg_male + geom_point(aes(x = chol))
# Status2 <- gg_male + geom_point(aes(x=stab.glu))
Status3 <- gg_male + geom_point(aes(x = hdl))
# Status4 <- gg_male + geom_point(aes(x=ratio)) Status5 <- gg_male +
# geom_boxplot(aes(x=location))
Status6 <- gg_male + geom_point(aes(x = age))
# Status7 <- gg_male + geom_col(aes(x=gender))
Status8 <- gg_male + geom_point(aes(x = height))
Status9 <- gg_male + geom_point(aes(x = weight))
# Status10 <- gg_male + geom_point(aes(x=frame))
Status11 <- gg_male + geom_point(aes(x = bp.1s))
Status12 <- gg_male + geom_point(aes(x = bp.1d))
# Status13 <- gg_male + geom_point(aes(x=waist)) Status14 <- gg_male +
# geom_point(aes(x=hip))
Status15 <- gg_male + geom_point(aes(x = time.ppn))

grid.arrange(Status1, Status3, Status6, Status8, Status9, Status11, Status12, Status15,
```

```
    nrow = 2, ncol = 4, top = "Status Variable Comparison")

# grid.arrange(Status1,Status2,Status3,Status4,Status5,
# Status6,Status7,Status8,Status9,Status10,
# Status11,Status12,Status13,Status14,Status15, nrow=3, ncol=5)

diabetes$ratio = NULL
diabetes_cor <- cor(subset(diabetes, select = -c(location, gender, frame, status)),
    use = "complete.obs")
corrplot(diabetes_cor, method = "color", type = "lower", addCoef.col = "black", diag = FALSE,
    main = "\nCorrelation Plot")
glm_full_probit <- glm(status ~ (. - glyhb), family = binomial(link = "probit"),
    data = diabetes)
# VIF(glm_full_probit)
data("diabetes", package = "faraway")
diabetes <- diabetes
diabetes <- subset(diabetes, select = -c(id))

# Remove 2nd Reading
diabetes <- subset(diabetes, select = -c(bp.2s, bp.2d))

# Remove Wasit/Hip/Ratio/Stab Glucose
diabetes$waist = NULL
diabetes$hip = NULL
diabetes$ratio = NULL
diabetes$stab.glu = NULL

# Remove remaining NA rows
diabetes <- diabetes[complete.cases(diabetes), ]

# Dichotomize GlyHB, Rename
diabetes$status <- diabetes$glyhb
diabetes[which(diabetes[, "glyhb"] <= 7), "status"] = 0
diabetes[which(diabetes[, "glyhb"] > 7), "status"] = 1

# Final Summary
summary(diabetes)

glm_full_probit <- glm(status ~ (. - glyhb), family = binomial(link = "probit"),
    data = diabetes)
glm_full_cloglog <- glm(status ~ (. - glyhb), family = binomial(link = "cloglog"),
    data = diabetes)

# summary(glm_full_probit) summary(glm_full_cloglog)

glm_nested_cloglog <- glm(status ~ chol + hdl + age + weight, family = binomial(link = "cloglog"),
    data = diabetes)

# summary(glm_nested_cloglog)

LL_full <- logLik(glm_full_cloglog)
LL_part <- logLik(glm_nested_cloglog)
```

```r
LL_teststat <- 2 * (as.numeric(LL_full) - as.numeric(LL_part))

LL_pval <- pchisq(LL_teststat, df = (12 - 4), lower.tail = FALSE)


gam_full_cloglog <- gam(status ~ s(chol) + s(hdl) + s(age) + s(height) + s(weight) +
    s(bp.1s) + s(bp.1d) + s(time.ppn) + location + gender + frame, family = binomial(link = "cloglog"),
    data = diabetes)

# summary(gam_full_cloglog)

###
gam_smooth_nested_cloglog <- gam(status ~ s(chol) + s(hdl) + s(age) + s(weight) +
    s(time.ppn), family = binomial(link = "cloglog"), data = diabetes)
par(mfrow = c(1, 5))
plot(gam_smooth_nested_cloglog, se = T)

gam_nosmooth_nested_cloglog <- gam(status ~ chol + s(hdl) + s(age) + s(time.ppn) +
    weight, family = binomial(link = "cloglog"), data = diabetes)
# summary(gam_nosmooth_nested_cloglog)

###

gam_final_cloglog <- gam(status ~ chol + hdl + s(age) + weight, family = binomial(link = "cloglog"),
    data = diabetes)
# summary(gam_final_cloglog)

LL_full_gam <- logLik(gam_full_cloglog)
LL_part_gam <- logLik(gam_final_cloglog)

LL_teststat_gam <- 2 * (as.numeric(LL_full_gam) - as.numeric(LL_part_gam))

LL_pval_gam <- pchisq(LL_teststat_gam, df = (20.07 - 6.11), lower.tail = FALSE)

# QQ PLOTS
set.seed(1)
par(mfrow = c(1, 2))
qqnorm(qresid(glm_nested_cloglog), main = "GLM QQ-Plot")
qqline(qresid(glm_nested_cloglog))
qqnorm(qresid(gam_final_cloglog), main = "GAM QQ-Plot")
qqline(qresid(gam_final_cloglog))
```