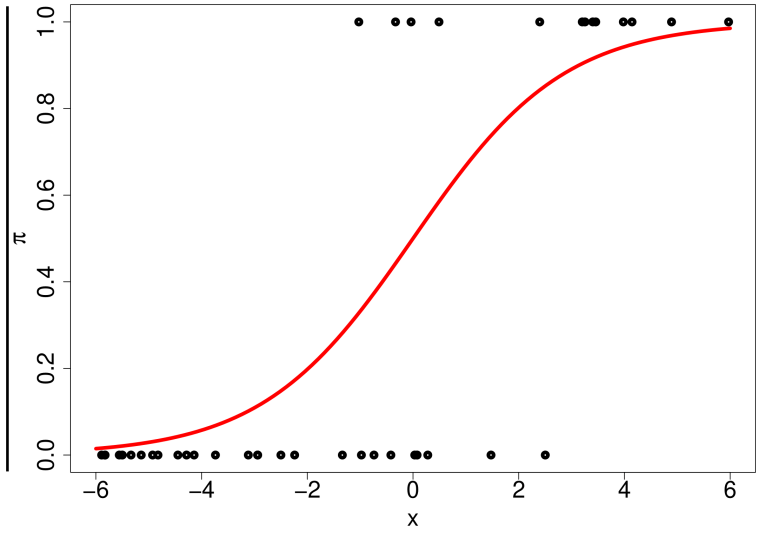
Goodness-of-Fit Tests for Binary GLMs in the Case of No/Few Replicates

Nikola Surjanovic
301289816
nsurjano@sfu.ca

Peter Tea
301392719
peter_tea@sfu.ca



1 Introduction

Formally checking model assumptions sounds like a reasonable idea, but the problem is more complex than it appears to be at first sight. There are several difficulties with making use of formal goodness-of-fit (GOF) tests. Without going into too many details, our first issue stems from the "dimensions" of the null and alternative hypotheses. Oftentimes, the null hypothesis is of lower "dimension" than the alternative hypothesis, and we know that with a sufficiently large sample size we will reject the null hypothesis with great probability. For example, an application of the Cramér–von Mises or Kolmogorov-Smirnov test for normality, given a moderate sample size, will likely result in rejecting the null hypothesis of normality. Another issue is the theoretical validity of various GOF tests under certain circumstances. Not every test is able to perform adequately in all scenarios, and we focus on this latter issue throughout our summary. Our view on the first issue for this report is that we will be using such GOF tests as aids, rather than blindly applying them. However, this approach does not imply that these tests should not have a solid theoretical basis.

In our context, we will deal with GOF tests for binary generalized linear models (GLMs) with no or few replicates. We now define these concepts briefly. Suppose that we have J tuples, (x_i, n_i, Y_i), $i = 1, \ldots, J$. The x_i's represent observed data (covariates) and lie in $\mathbb{R}^p$, whereas the Y_i represent unobserved (random) responses that are related in some way to the covariates and are in $\{0, 1, \ldots, n_i\}$. If we assume that

$$Y_i | X_i = x_i \sim Bin(n_i, \pi_i) \quad i = 1, \ldots, J,$$

with

$$q(\pi_i) = \beta^\top x_i,$$

for some monotone and continuously differentiable link function q (specified in advance), and a p-dimensional vector β, we then have a GLM with binary/binomial responses. Of course, one might not be willing to refer to the above model as a *binary* GLM, due to binomial responses. However, binomial random variables are simply sums of independent Bernoulli random variables, and the model can therefore be rewritten with binary responses. The above model definition is useful, however, because it allows for the illustration of two very important cases: *no and few replicates*. We say that we have a binary GLM with *no replicates* when each $n_i = 1$, i.e., $n \equiv \sum_i n_i = J$. Such a GLM is shown in Figure 1. On the other hand, a binary GLM with *few replicates* is one in which each n_i is relatively small. Few or no replicates can occur, for example, when we have a continuous explanatory variable present in our model, or a discrete variable that can take on infinitely many values.

The title page is based on a template by Harish Kumar and "azetina" on StackExchange. (Nikola reuses the same title page on different projects.)

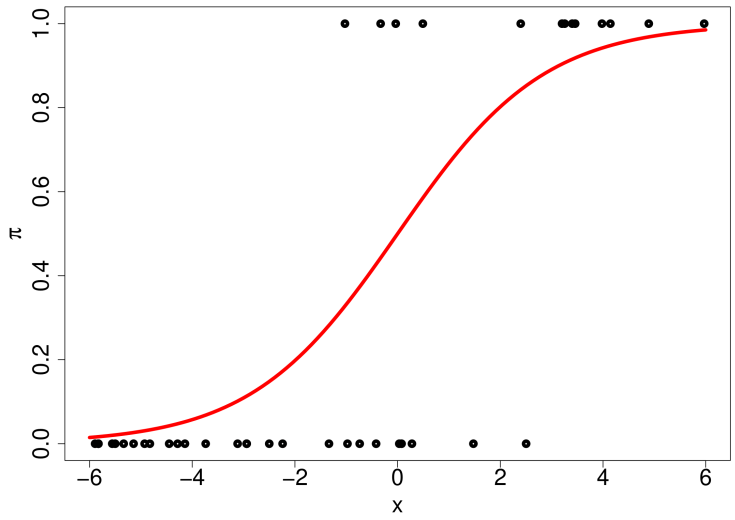


Figure 1: An example of a binary GLM (logit link) with no replicates. Black dots represent observed covariates and responses, (x_i, y_i)'s, while the red curve is the true underlying regression function.

The saturated model is the model that fits a separate mean for each unique covariate pattern, x_i. We previously saw that the Pearson chi-squared test can be used to test the following:

H_0 : Proposed model fits as well as the saturated model,

vs.

H_A : Proposed model does not fit as well as the saturated model.

The corresponding Pearson chi-squared test statistic is

$$X^2 = \sum_{i=1}^{J} \frac{(Y_i - n_i\hat{\pi}_i)^2}{n_i\hat{\pi}_i(1-\hat{\pi}_i)}, \tag{1}$$

where the $\hat{\pi}_i$'s are the fitted values from the proposed model. Under the null hypothesis, we have that

$$X^2 \xrightarrow{d} \chi^2_{J-p},$$

provided that:

1. J is fixed and does not grow with $n \equiv \sum_i n_i$, and
2. As $n \to \infty$, each $n_i \to \infty$.

Sometimes the second condition is phrased as "each n_i is sufficiently large", in order to allow for finite sample size approximations.

From the first condition above, we see that if J increases with n, then the number of replicates within each unique covariate pattern will likely be small, i.e., the second condition is likely violated. As mentioned before, a common example of when this condition may be violated is when we have a continuous covariate present in our model. Focusing on the extreme case in which each $n_i = 1$ and $J = n$, we might falsely believe that

$$X^2 \xrightarrow{d} \chi^2_{n-p} \quad (?)$$

However, this statement is clearly nonsensical due to the use of n on the right side of the convergence statement. Therefore, something needs to be done before we can use the regular Pearson chi-squared test in the case of no or few replicates. We will illustrate throughout this report which GOF tests can be used in such circumstances.

2 Test Statistics

2.1 Osius-Rojek Test

Considering the Pearson chi-squared test statistic, X^2, in the case of no replicates, we just saw that the limiting distribution of this statistic under the null hypothesis is *not* χ^2_{n-p}. The question remains: what can we do about this test statistic? Fortunately, this question was addressed by Osius and Rojek (1992).

We begin by noting that, in this case,

$$X^2 = \sum_{i=1}^{n} \frac{(Y_i - \hat{\pi}_i)^2}{\hat{\pi}_i(1 - \hat{\pi}_i)}.$$

This is clearly a sum of independent terms. Note that the terms are not identically distributed if we are working in the framework of non-stochastic x_i (for example, in experimental design). Regardless, with proper standardization, a central limit theorem such as the Lyapunov or Lindeberg central limit theorem can be invoked. In fact, even in scenarios in which we have few or many replicates, X^2 as given by (1) can be standardized to obtain that

$$\frac{X^2 - \hat{\mu}}{\hat{\sigma}} \xrightarrow{d} N(0, 1),$$

for some $\hat{\mu}, \hat{\sigma}^2$, under appropriate regularity conditions (see Osius and Rojek, 1992).

What are $\hat{\mu}$ and $\hat{\sigma}^2$, then? It turns out that, regardless of the number of replicates,

$$\hat{\mu} = J = n.$$

With no replicates, $\hat{\sigma}^2$ can be obtained through a residual sum of squares based on output from a weighted least squares regression on a transformed version of the original data. In the case of some replicates, the formula for $\hat{\sigma}^2$ is similar, except that an additional term is added.

What is even more impressive is that Osius and Rojek (1992) build upon the work of Read and Cressie (1988), essentially developing infinitely many test statistics, indexed by a parameter λ. For example, when $\lambda = 1$, a standardized Pearson test statistic is obtained. Setting $\lambda = 0$ yields a standardized deviance test statistic. This work is based on a generalized distance between the "observed" and expected counts, Y_i and $n_i\hat{\pi}_i$, respectively. Namely, defining

$$a_\lambda(Y_i, n_i\hat{\pi}_i) = \frac{2Y_i}{\lambda(\lambda+1)} \cdot \left[\left(\frac{Y_i}{n_i\hat{\pi}_i}\right)^\lambda - 1\right] - \frac{2}{\lambda+1}(Y_i - n_i\hat{\pi}_i),$$

we see that

$$\begin{aligned} a_1(Y_i, n_i\hat{\pi}_i) &= \frac{2Y_i}{1(1+1)} \cdot \left[\left(\frac{Y_i}{n_i\hat{\pi}_i}\right)^1 - 1\right] - \frac{2}{1+1}(Y_i - n_i\hat{\pi}_i) \\ &= \frac{(Y_i - n_i\hat{\pi}_i)^2}{n_i\hat{\pi}_i}, \end{aligned}$$

after some algebraic manipulation. As will be discussed again in the next subsection on the Hosmer-Lemeshow test,

$$a_1(Y_i, n_i\hat{\pi}_i) + a_1((n_i - Y_i), n_i(1-\hat{\pi}_i)) = \frac{(Y_i - n_i\hat{\pi}_i)^2}{n_i\hat{\pi}_i(1-\hat{\pi}_i)}.$$

We can therefore define a general statistic,

$$SD_\lambda = a_\lambda(Y_i, n_i\hat{\pi}_i) + a_\lambda((n_i - Y_i), n_i(1-\hat{\pi}_i)), \tag{2}$$

and its approximate standardized version

$$T_\lambda = \frac{SD_\lambda - \hat{\mu}_\lambda}{\hat{\sigma}_\lambda}. \tag{3}$$

This new statistic has the property that

$$T_\lambda \xrightarrow{d} N(0,1),$$

under certain conditions given by Osius and Rojek (1992).

Unfortunately, the authors give simple formulae for $\hat{\mu}_\lambda$ and $\hat{\sigma}_\lambda$ only for some common values of λ. Also, it is not clear how to generalize their work to GLMs containing a non-binomial response. Osius and Rojek (1992) develop results for

product-multinomial models, so Poisson regression, for example, would be difficult (likely impossible) to incorporate into their framework.

Finally, a discussion should be offered on calculating p-values for the Osius-Rojek test. We know how to maintain the type 1 error rate from the limiting distribution of the test statistic, but an appropriate calculation of the p-value will ensure high power and avoid nonsensical conclusions. For the Osius-Rojek standardized Pearson chi-squared test ($\lambda = 1$) and $J = n$ (no replicates), we should conduct a *two-tailed* test. In other words, we should reject for both large and small values of the test statistic. This might come as a surprise, because of the one-tailed nature of the Pearson chi-squared test (i.e., where we reject only for large values of the test statistic). A relatively detailed explanation is given by Osius and Rojek (1992).

2.2 Hosmer-Lemeshow Test

The Hosmer-Lemeshow (HL) GOF test was first proposed in 1980, as seen in Hosmer and Lemeshow (1980). This test was one of the first GOF tests designed specifically for cases when we have no or few replicates. To facilitate the illustration of the HL test, we first present a contingency table grouped by covariate pattern.

	1	2	$\cdots$	J	
y = 0	1	0			
y = 1	0	2			

Here, the contingency table is arranged in dimension (2 x J), such that the 2 rows represent the binary response and the J columns represent the number of covariate groups, where $J \leq n$. In the case of few or no replicates, the expected cell counts in the above table may not be sufficiently large to justify the use of a Pearson chi-squared goodness of fit test.

To bypass the issue of no or few replicates, the HL procedure proposes grouping observations by their respective fitted values under the proposed model, rather than by covariate pattern. In the binary GLM case, note that the proposed model will provide fitted values for each covariate pattern. Under the HL construct, observations similar in fitted probability values will be partitioned into the same subgroups. There exist different strategies to systematically partition the observational units into meaningful subgroups, and we present them in section 2.2.1.

The partitioning of observational units into g subgroups effectively condenses the dimension of the contingency table. In fact, under this grouping scheme, the original (2 x J) contingency table collapses into a (2 x g) table, where $g \leq J$.

	1	2	$\cdots$	g	
y = 0	1	0			
y = 1	0	2			

With the grouping idea proposed by Hosmer and Lemeshow (1980), the corresponding collapsed contingency table may now have expected cell counts that are more likely to be sufficiently large to justify the use of the chi-squared test for goodness-of-fit. Below, we present the Hosmer-Lemeshow test statistic for GOF from Hosmer and Lemeshow (1980) and Hosmer (2013). The Hosmer-Lemeshow test statistic can be calculated as follows:

$$\widehat{C}_g = \sum_{k=1}^{g} \left[\frac{(O_{1k} - \hat{E}_{1k})^2}{\hat{E}_{1k}} + \frac{(O_{0k} - \hat{E}_{0k})^2}{\hat{E}_{0k}} \right], \tag{4}$$

where

$$O_{1k} = \sum_{I_k} Y_i \quad \text{(The number of successes in the } k\text{th group.)}$$

$$O_{0k} = \sum_{I_k} (n_i - Y_i) \quad \text{(The number of failures in the } k\text{th group.)}$$

$$\hat{E}_{1k} = \sum_{I_k} n_i \hat{\pi}_i \quad \text{(The expected number of successes in the } k\text{th group.)}$$

$$\hat{E}_{0k} = \sum_{I_k} n_i (1 - \hat{\pi}_i) \quad \text{(The expected number of failures in the } k\text{th group.)}$$

I_k is the the set of indices for the obersvations in the kth subgroup.

It can be shown that the HL statistic can also be written as:

$$\widehat{C}_g = \sum_{k=1}^{g} \frac{(O_{1k} - n'_k \bar{\pi}_k)^2}{n'_k \bar{\pi}_k (1 - \bar{\pi}_k)}, \tag{5}$$

where

$$\bar{\pi}_k = \frac{1}{n'_k} \sum_{I_k} n_i \hat{\pi}_i,$$

and n'_k is the number of units in the kth subgroup.

Under the formulation above, we can more easily see the resemblance of this procedure to the chi-squared GOF test. Here, the expected number of successes in each kth group is estimated by multiplying the mean of the fitted values within that kth group and the kth group size.

Through a simulation study conducted by Hosmer and Lemeshow (1980), it has been found that when $J = n$ (i.e., when there are no replicates), and when the fitted logistic model is the correct model, then the distribution of the $\hat{C}_g$ statistic is well approximated by the χ^2_{g-2} distribution. Similarly, when $J \approx n$ (i.e. when there are few replicates), then the asymptotic result holds as well.

2.2.1 Strategies to Split Units Into Meaningful Subgroups

The demarcation of the observational units into g subgroups can be done in many ways. We present 2 main strategies proposed by Hosmer and Lemeshow:

1. Group observational units based on percentiles of estimated probabilities
2. Group observational units based on fixed values of the estimated probabilities

How do the two grouping strategies above work? Consider an example where we set 10 groups (i.e., $g = 10$), letting n be the total number of units. Under strategy 1, we would split the estimated probabilities into group sizes of $n_g = n/10$. Here, n_1 would contain units with the smallest estimated probabilities and n_{10} contains the units with the largest estimated probabilities. Note that each group contains approximately 10% of the sampling units. This grouping strategy is sometimes referred to as "deciles-of-risk", where observations are ranked according to their predicted probability and put into $g = 10$ approximately equal sized groups (see Canary et al., 2017). Under strategy 2, we would define 10 fixed cut-points that divide the units into 10 groups. Here, the first group would contain the units with estimated probabilities less than or equal to 0.1, while the 10th group would contain the units with estimated probabilities greater than 0.9. The grouping method based on percentiles (i.e. Strategy I), is the more widely used grouping method when implementing the Hosmer-Lemeshow test.

2.2.2 Drawbacks of the Hosmer-Lemeshow Test

One issue with the HL test is the lack of guidance in choosing an appropriate number of groups, g. The choice of g is subjective and can unfortunately impact the results of the HL test. Paul Allison, a statistician from the University of Pennsylvania, wrote a post titled "Why I Don't Trust the Hosmer-Lemeshow Test for Logistic Regression"[1], which highlights the issue of HL group sizes. He writes that there exist situations where "if one changes g, sometimes one obtains a quite different p-value, such that with one choice of g we might conclude our model does not fit well, yet with another we conclude there is no evidence of poor fit". Hence, one main criticism of the HL

[1] https://statisticalhorizons.com/hosmer-lemeshow

test is that the results may rely on an arbitrary choice of number of groups. This unfortunately is a drawback of the HL test that has yet to be rectified.

2.3 C_n Statistic

Recall that the sum of squared Pearson residuals for binary responses (with no replicates) is as follows:

$$\sum_{i=1}^{n} \frac{(Y_i - \hat{\pi}_i)^2}{\hat{\pi}_i (1 - \hat{\pi}_i)}.$$

The above formulation has been shown to asymptotically follow a standard normal distribution when properly standardized (see Windmeijer, 1989). We may consider the above squared Pearson residuals as a GOF test statistic. However, it has been noted that this statistic's variance is unstable with extreme values of π_i. One possible variant of the sum of squared standardized residuals with stabilized variance is presented below as the C_n test statistic by Chen et al. (2018):

$$C_n = \sum_{i=1}^{n} \frac{Y_i - \hat{\pi}_i}{\sqrt{\hat{\pi}_i (1 - \hat{\pi}_i)}} \tag{6}$$

Under some regularity conditions,

$$\frac{n^{-\frac{1}{2}} C_n}{\hat{\sigma}_n} \xrightarrow{d} N(0, 1).$$

One main regularity assumption required to reach the distributional result above is the assumption that "there exists a finite number M such that

$$\|x_i\| \leq M, \forall i \in \mathbb{N} \text{ and } \lim_{n \to \infty} \sum_i x_i x_i^\top$$

is a finite non-singular matrix." Hence, we can use the the above result to conduct a 2-sided goodness of fit test. Unfortunately, due to the regularity assumption made, one main drawback from this test is that it can only be applied in the case where there are no replicates.

2.4 Stukel Test

The test proposed by Stukel (1988) is quite different from the other tests that have been or will be considered in this summary. The idea behind the Stukel test is

to first define a family of functions, $\{h_\alpha\}_{\alpha\in\mathbb{R}^2}$, indexed by $\alpha = (\alpha_1, \alpha_2)^\top$. These two parameters control the left and right tails of the inverses of the resulting link functions, respectively. If $\alpha = (0,0)^\top$, the resulting link function is identical to the usual logit link. We denote the linear predictor, $\beta^\top x$, by η, and then define

$$\pi_\alpha(\eta) = \frac{\exp(h_\alpha(\eta))}{1+\exp(h_\alpha(\eta))}. \tag{7}$$

If we replace q, from before, with π_α, we obtain a model similar to the GLM introduced at the beginning of this report, except that we assume that the probability of "success" is of the form given in (7), for some α, β. This model—which is a regular GLM if α is fixed—is fit, finding the optimal values of α and β. If α differs significantly from $(0,0)^\top$, then the logit link is rejected. Therefore, this is a test that assesses the link function directly.

Clearly, the link function is not the only model assumption behind a GLM. However, if we imagine omitting a variable that is correlated with another variable from our GLM, there should be a change in the estimated regression function. For example, a quadratic term that is added or removed from a model can noticeably change the shape of the prediction curve. Therefore, a test of the link function might also be able to capture information about the linear predictor and whether or not it is correctly specified.

The family of functions that Stukel (1988) defined is now described. For $\eta \geq 0$,

$$h_\alpha(\eta) = \begin{cases} 1/\alpha_1(\exp(\alpha_1|\eta|)-1), & \alpha_1 > 0 \\ \eta, & \alpha_1 = 0 \\ -\alpha_1\log(1-\alpha_1|\eta|), & \alpha_1 < 0 \end{cases}.$$

Similarly, for $\eta \leq 0$,

$$h_\alpha(\eta) = \begin{cases} -1/\alpha_2(\exp(\alpha_2|\eta|)-1) & \alpha_2 > 0 \\ \eta & \alpha_2 = 0 \\ \alpha_2\log(1-\alpha_2|\eta|) & \alpha_2 < 0 \end{cases}.$$

Examples of some of the resulting link functions are displayed in Figure 2.

Another interesting comment regarding the test by Stukel, is that the family of link functions indexed by α is quite rich. We have already mentioned that when $\alpha = (0,0)^\top$ we obtain the regular logit link. However, it is also possible to obtain approximations to other familiar link functions. For example, if $\alpha = (0.165, 0.165)^\top$, one gets a link function very similar to the probit link. On the other hand, setting $\alpha = (0.62, -0.037)^\top$ yields a link function that is close to the complementary log-log link. This is, in fact, a consequence of the design of the family of link functions. For

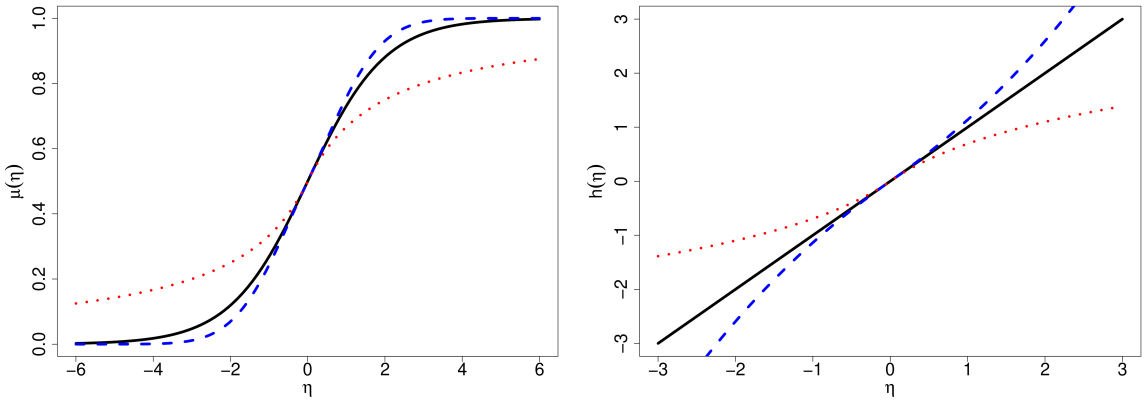


Figure 2: Left: Various link functions, as defined by Stukel (1988). This is a plot of the inverse of the link functions, as a function of the linear predictor, η. The black curve is the inverse of the regular logit link (i.e., the CDF of the logistic distribution). The blue dashed curve corresponds to $\alpha = (0.25, 0.25)^\top$, while the red dotted curve corresponds to $\alpha = (-1, -1)^\top$. (These values are the same as those used in the plot in Stukel's paper.) Right: The same graph as on the left, except that the logit function is applied to the inverses of the link functions. The black curve therefore corresponds to the logit link.

any given α, $\pi_\alpha(x)$ is the cumulative distribution function (CDF) of some random variable whose density is given by $\pi'_\alpha(x)$. By calculating the first several moments of this corresponding random variable, one can solve for the values of α that approximate certain well-known distributions to the first few moments. Although there are only two free link function parameters, α_1 and α_2, the standard normal and extreme minimum distributions can be approximated to the first four moments.

Now that we have a family of link functions, two questions remain: how do we fit the model, and how do we test, for example, whether or not $\alpha = (0, 0)^\top$? A naive way of answering the first question is to select a grid of values of α, and for each value of α to fit the appropriate GLM to find the MLEs for β. Then, the values of α and β that maximize the likelihood should be used. However, Stukel (1988) proposed a more clever approach. She suggests the use of the "delta algorithm" (see Jørgensen, 1984), which is similar to the iteratively reweighted least squares algorithm, except that the design matrix changes at each iteration, and some other small modifications are introduced. For testing whether $\alpha = (0, 0)^\top$, one can make use of a score test. If l is the log-likelihood, and $(\partial l/\partial\alpha_1, \partial l/\partial\alpha_2)$ are the elements of the score corresponding to α, evaluated at $\alpha = (0, 0)^\top$ and $\beta = \hat{\beta}_0$, the β that maximizes the log-likelihood at $\alpha = 0$, we can then test whether

$$(\partial l/\partial\alpha_1, \partial l/\partial\alpha_2)A^{-1}(\partial l/\partial\alpha_1, \partial l/\partial\alpha_2)^\top$$

is large compared to a χ^2_2 distribution, for an appropriate 2×2 matrix A (see Stukel, 1988). Of course, we can also likely conduct a likelihood ratio test as an alternative.

Although it doesn't seem that Stukel (1988) mentions this directly, it should also be possible to test for values of α other than $\alpha = (0, 0)^\top$. For example, we might want to test whether the probit link is appropriate, if we have reason to believe that this link function describes the phenomenon more accurately. In this case, we would test $\alpha = (0.165, 0.165)$.

2.5 Other Tests

There are quite a few other GOF tests for binary GLMs with no or few replicates. Some of these are summarized by Chen et al. (2018) and Hosmer et al. (1997). However, there is one more test worth mentioning: the information matrix test statistic.

The information matrix test statistic, introduced by White (1982), is useful because of its generality. It is not limited to GLMs, and can be used with almost any model that is fit using maximum likelihood estimation. The idea is to compare two estimates of the Fisher information matrix. We change notation temporarily to illustrate that this test is not restricted to GLMs. Suppose that we have iid (unobserved) data $X_1, \ldots, X_n$, having the same distribution as X, which is parameterized by $\theta \in \mathbb{R}^d$. Denoting the log-likelihood by $\log(f(X; \theta))$, under some regularity conditions, we have that

$$\mathrm{E}\left[\left(\frac{\partial}{\partial\theta}\log(f(X;\theta))\right)\left(\frac{\partial}{\partial\theta}\log(f(X;\theta))\right)^\top\right] = -\,\mathrm{E}\left[\left(\frac{\partial^2}{\partial\theta\partial\theta}\log(f(X;\theta))\right)\right].$$

That is, the outer product form and Hessian form of the information matrix should be equivalent under some conditions. However, these forms (and their estimates) differ *if the model is misspecified*. Denoting the first matrix by $A_1(\theta)$ and the *negative* of the second matrix by $A_2(\theta)$, an idea is to check whether or not

$$n(\hat{A}_1(\hat{\theta}) + \hat{A}_2(\hat{\theta}))C^{-1}(\hat{A}_1(\hat{\theta}) + \hat{A}_2(\hat{\theta}))^\top$$

is large relative to a χ^2_d distribution (note the use of an addition sign because off the definition of A_2), where $\hat{\theta}$ is the MLE of θ, $\hat{A}_1, \hat{A}_2$ are estimates of A_1, A_2 when the expectation can not be evaluated, and C is an appropriate matrix. White (1982) explains how to properly construct a test statistic from these two estimates of the Fisher information matrix, so that the proposed test statistic converges in distribution to a chi-squared random variable with d degrees of freedom.

3 Application in R

We illustrate the use of some of the GOF tests presented in this report, on a simple logistic model applied in tennis. We first scrape tennis data from Jeff Sackmann's GitHub repository detailing Novak Djokovic's serve speed (continuous variable), and outcome of service points (binary response). We fit a binary logistic regression model where the response is the outcome of the service point, and the covariate is the continuous serve speed variable.

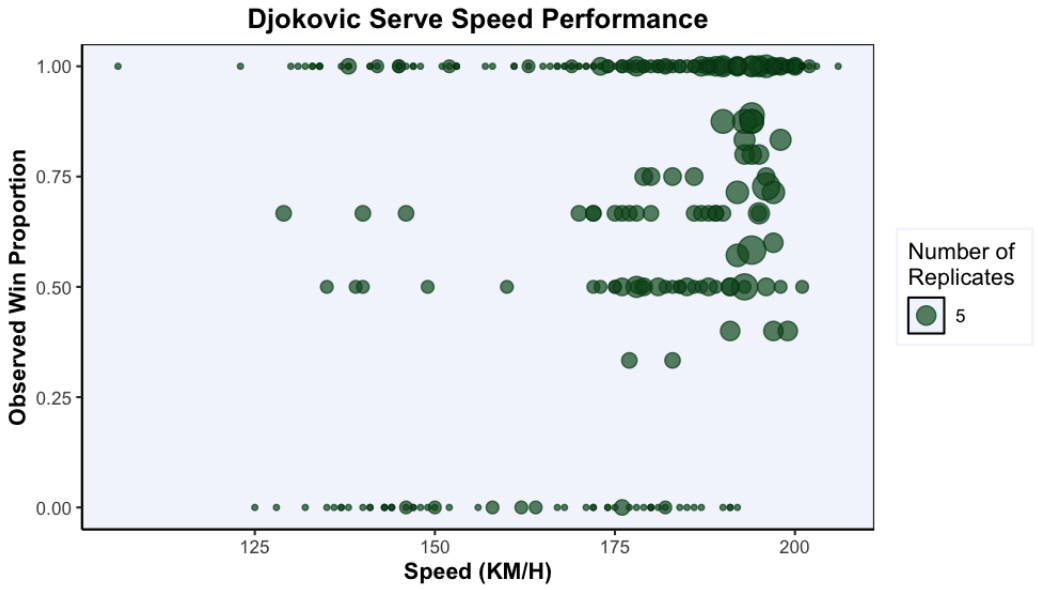


Figure 3: Novak Djokovic's proportion of points won for each observed serve speed during the Australian Open 2020 tournament.

From Figure 3, we show that the continuous serve speed variable has many covariate patterns with no or few replicates. This is illustrated by considering the size aesthetic representing the number of replicates for each covariate pattern. Next, we fit a simple logistic regression model with point outcome (won point or not) as the binary response variable and serve speed as the continuous predictor variable. We present the results obtained from applying some of the GOF tests described throughout the report on our fitted serve speed model.

From Table 1, we make the following remarks. Firstly, the Hosmer-Lemeshow test provides different results when equipped with slightly different group sizes, g. In this case, we would end up with the same conclusion (assuming a significance level of 0.05). Secondly, the Osius-Rojek test agrees with the aforementioned Hosmer-Lemeshow tests in that it also does not reject H_0, and hence does not conclude that there is evidence of lack of fit. Lastly, the Stukel test is the only goodness of fit test

Goodness of Fit Test	Test Statistic	p-value
Hosmer-Lemeshow ($g = 10$)	7.12	0.5
Hosmer-Lemeshow ($g = 11$)	11.4	0.25
Osius-Rojek	0.05	0.96
Stukel	6.35	0.042

Table 1: Goodness-of-fit tests applied on a serve speed logistic model.

here with a statistically significant p-value at the level of 0.05. Hence, if going by the results from the Stukel test, we would conclude that there exists evidence that the fitted speed model lacks goodness of fit.

With our serve speed model, we don't know a priori whether this fitted model was the correct model. Hence, it is difficult to examine the performance of the competing GOF tests. To rectify this issue, we fit a second serve speed logistic model, but now without an intercept term. This second serve speed model should have a poor fit, and we would like our GOF tests to identify this lack of fit. We present the results obtained from applying the goodness-of-fit tests, on our poor-fitting serve speed model.

Goodness of Fit Test	Test Statistic	p-value
Hosmer-Lemeshow (g=10)	16.2	0.04
Hosmer-Lemeshow (g=11)	19.8	0.02
Osius-Rojek	3.9	< 0.01
Stukel	5.96	0.015

Table 2: Goodness-of-fit tests applied on a poor-fitting serve speed logistic model without intercept.

From Table 2, reassuringly we see that all 4 goodness-of-fit tests are able to detect that our second serve speed model has a poor fit to the data. Unfortunately, there appears to be little separation between the performances of these GOF tests. In the next section, we look at a simulation study that compares the performances of these GOF tests in greater detail.

4 Comparison of Different GOF Tests

Given the high volume of different GOF tests presented in this report (and the bevy of other GOF tests that exists, but were not mentioned here), it is important to explore how the competing tests perform relative to one another. We present a summary of a simulation study conducted in Chen et al. (2018) to illustrate the performances of some of the competing GOF tests. In particular, the simulation study examines the relative performances of the following tests: C_n, Hosmer-Lemeshow ($g = 10$), Hosmer-Lemeshow ($g = 20$), Stukel, and IM (Information Matrix).

The structure of the simulated data was inspired from a real dataset from an epidemiological study conducted on participants of the 2002 Boston Marathon. The binary response of interest here was whether runners had developed hyponatremia. The covariates were a mix of both categorical and continuous variables.

In the simulation study, the authors consider three constructs to generate a binary response outcome:

- Model 1.0: The true null model
- Model 1.1: An alternative model with an added quadratic covariate
- Model 1.2: An alternative model with an added quadratic covariate along with different coefficient values

For each of the 3 models above, the authors then look at the proportion of H_0 rejections among the competing GOF tests. A sample size of 445 was used in conjunction with 10,000 replications. Results from this simulation study can be found in Table 2 of Chen et al. (2018).

From the simulation, the authors found that when the fitted model is correct (i.e., when H_0 is true, model 1.0), the C_n test consistently rejects H_0 near the nominal level. The other competing tests appear to reject the null at a proportion that is higher than the nominal level of 0.05. The Stukel test and the HL test with $g = 20$ appear to have the most inflated type I error. Furthermore, the authors found that when the model is incorrectly specified (models 1.1 and 1.2), the C_n statistic is able to detect lack of fit nearly as well as the other competing statistics. From the simulation study, the authors argue that the C_n test statistic strikes a good balance between power and an appropriate type I error rate. In terms of overall performance, the C_n statistic is generally stable and performs as good as or better than other GOF statistics.

5 Discussion and Summary

Throughout this report we discussed why having few or no replicates in binary GLMs is an issue, introduced various GOF tests to handle this problem, and assessed the

performance of these tests through a simulation study conducted by Chen et al. (2018) as well as our own example on tennis data. Of course, these GOF tests should also be accompanied by various diagnostic checks, such as residual plots (although these diagnostic plots also do not behave well when there are few or no replicates, so techniques such as grouping of the data are likely necessary). Since we will often be using these tests as an aid, we might consider applying as many GOF tests as possible. However, in this scenario we need to be careful not to select the smallest p-value or combine the p-values in some way without proper theoretical justification.

We now present a brief summary of the pros and cons of each of the tests presented. The Osius-Rojek test is easy to compute, and not restricted to the logit link. However, the test does not generalize well to non-binary response GLMs. For example, one would not be able to use the Osius-Rojek test for Poisson GLMs. The Hosmer-Lemeshow test is also simple and intuitive, much like the Osius-Rojek test, but the test statistic and p-value can change depending on the somewhat arbitrary choice of g. On the other hand, this test can be generalized to other GLMs with a proper modification (a work in progress). The C_n test statistic is simple and relatively straightforward to implement, but only works for GLMs with no replicates. An interesting test, the Stukel test assesses appropriateness of the link function directly, but the family of link functions is constrained to those with $(0, 1)$ as the domain. Therefore, such link functions are most suitable for use with binary regression models, and we again see a restriction of the test to a certain class of GLMs. Finally, the information matrix test statistic is very general. The results from the somewhat limited simulation study seem to suggest that this test has moderate power, however.

GOF testing has been around in statistics for a long time, and usually has quite a mathematical flavour. Interesting directions for future research might involve addressing the first issue brought up at the beginning of this report. How can we ensure that our null hypothesis is "large enough" so that we are not rejecting the null hypothesis simply due to a large sample size? A Bayesian approach with a definition of distances between models might work, as explored by Swartz (1999) to a certain extent. For example, if we wish to test the link function, we can perhaps define a distance between link functions (via the sup norm, for example), and test whether or not our link function lies within a certain range of functions. In a Bayesian context, with a suitable prior on the link functions, we might even be able to obtain the probability that the null hypothesis is "correct". With this setup, we could also theoretically obtain credible intervals suggesting where the true model might be located in the space of possible models.

6 Contributions

For this report, we divided the work as follows:

Peter

- Part of the introduction
- Section on the Hosmer-Lemeshow test
- Section on the C_n test statistic
- Scraping tennis data for the R demo and code for the application, as well as the section for the "Application in R"
- Summary of the simulation study from Chen et al. (2018) that compares various GOF tests
- Review of overall report and editing

Nikola

- Part of the introduction
- Section on the Osius-Rojek test
- Section on the Stukel test
- Description of other tests (information matrix test statistic and other references)
- Discussion and summary
- Wrote R code for the information matrix and C_n GOF tests (the Stukel, Osius-Rojek, and HL test functions were already written by Tom Loughin)
- Review of overall report and editing (mostly making sure that notation and formatting was the same across both halves, as well as doing grammar checks)

References

Jana D. Canary, Leigh Blizzard, Ronald P. Barry, David W. Hosmer, and Stephen J. Quinn. A comparison of the Hosmer-Lemeshow, Pigeon-Heyse, and Tsiatis goodness-of-fit tests for binary logistic regression under two grouping methods. *Communications in Statistics - Simulation and Computation*, 46(3):1871–1894, 2017. doi: 10.1080/03610918.2015.1017583.

Lu Chen, Yihan Sui, Chi Song, and Grzegorz A Rempala. The sum of standardized residuals: Goodness-of-fit test for binary response models. *Statistics in medicine*, 37(11):1932, 2018. ISSN 0277-6715.

David W. Hosmer. *Applied logistic regression / David W. Hosmer, Jr., Stanley Lemeshow, Rodney X. Sturdivant.* Wiley series in probability and statistics. Wiley, third edition. edition, 2013. ISBN 0470582472.

David W. Hosmer and Stanley Lemeshow. Goodness of fit tests for the multiple logistic regression model. *Communications in Statistics-Theory and Methods*, 9 (10):1043–1069, 1980.

David W. Hosmer, Trina Hosmer, Saskia Le Cessie, and Stanley Lemeshow. A comparison of goodness-of-fit tests for the logistic regression model. *Statistics in Medicine*, 16(9):965–980, 1997.

Bent Jørgensen. The delta algorithm and glim. *International Statistical Review/Revue Internationale de Statistique*, pages 283–300, 1984.

Gerhard Osius and Dieter Rojek. Normal goodness-of-fit tests for multinomial models with large degrees of freedom. *Journal of the American Statistical Association*, 87 (420):1145–1152, 1992.

T.R. Read and N.A. Cressie. Goodness-of-fit statistics for discrete multivariate data. 1988.

Thérèse A Stukel. Generalized logistic models. *Journal of the American Statistical Association*, 83(402):426–431, 1988.

Tim Swartz. Nonparametric goodness-of-fit. *Communications in Statistics-Theory and Methods*, 28(12):2821–2841, 1999.

Halbert White. Maximum likelihood estimation of misspecified models. *Econometrica: Journal of the Econometric Society*, pages 1–25, 1982.

F. Windmeijer. The asymptotic distribution of the sum of weighted squared residuals in binary choice models. (2243-2019-3567):16, 1989. doi: 10.22004/ag.econ.293145. URL http://ageconsearch.umn.edu/record/293145.

Combined R Code (Appendix)

Code for GOF Tests

The code for the first three tests can be found at http://www.chrisbilder.com/categorical/Chapter5/AllGOFTests.R.

```
#########################################################################
# NAME: Tom Loughin                                                     #
# DATE: 1-10-13                                                         #
# PURPOSE: Functions to compute Hosmer-Lemeshow, Osius-Rojek, and       #
#     Stukel goodness-of-fit tests                                      #
#                                                                       #
# NOTES:                                                                #
#########################################################################
# Single R file that contains all three goodness-of fit tests


# Adapted from program published by Ken Kleinman as Exmaple 8.8 on the SAS and R blog,
# sas-and-r.blogspot.ca
#  Assumes data are aggregated into Explanatory Variable Pattern form.

HLTest = function(obj, g) {
 # first, check to see if we fed in the right kind of object
 stopifnot(family(obj)$family == "binomial" && family(obj)$link == "logit")
 y = obj$model[[1]]
 trials = rep(1, times = nrow(obj$model))
 if(any(colnames(obj$model) == "(weights)"))
  trials <- obj$model[[ncol(obj$model)]]
 # the double bracket (above) gets the index of items within an object
 if (is.factor(y))
  y = as.numeric(y) == 2  # Converts 1-2 factor levels to logical 0/1 values
 yhat = obj$fitted.values
 interval = cut(yhat, quantile(yhat, 0:g/g), include.lowest = TRUE)
 # Creates factor with levels 1,2,...,g
 Y1 <- trials*y
 Y0 <- trials - Y1
 Y1hat <- trials*yhat
 Y0hat <- trials - Y1hat
 obs = xtabs(formula = cbind(Y0, Y1) ~ interval)
 expect = xtabs(formula = cbind(Y0hat, Y1hat) ~ interval)
 if (any(expect < 5))
  warning("Some expected counts are less than 5. Use smaller number of groups")
 pear <- (obs - expect)/sqrt(expect)
 chisq = sum(pear^2)
 P = 1 - pchisq(chisq, g - 2)
 # by returning an object of class "htest", the function will perform like the
 # built-in hypothesis tests
```

```
 return(structure(list(
  method = c(paste("Hosmer and Lemeshow goodness-of-fit test with", g, "bins", sep = " ")),
  data.name = deparse(substitute(obj)),
  statistic = c(X2 = chisq),
  parameter = c(df = g-2),
  p.value = P,
  pear.resid = pear,
  expect = expect,
  observed = obs
 ), class = 'htest'))
}

# Osius-Rojek test
# Based on description in Hosmer and Lemeshow (2000) p. 153.
# Assumes data are aggregated into Explanatory Variable Pattern form.

o.r.test = function(obj) {
 # first, check to see if we fed in the right kind of object
 stopifnot(family(obj)$family == "binomial" && family(obj)$link == "logit")
 mf <- obj$model
 trials = rep(1, times = nrow(mf))
 if(any(colnames(mf) == "(weights)"))
  trials <- mf[[ncol(mf)]]
 prop = mf[[1]]
 # the double bracket (above) gets the index of items within an object
 if (is.factor(prop))
  prop = as.numeric(prop) == 2  # Converts 1-2 factor levels to logical 0/1 values
 pi.hat = obj$fitted.values
 y <- trials*prop
 yhat <- trials*pi.hat
 nu <- yhat*(1-pi.hat)
 pearson <- sum((y - yhat)^2/nu)
 c = (1 - 2*pi.hat)/nu
 exclude <- c(1,which(colnames(mf) == "(weights)"))
 vars <- data.frame(c,mf[,-exclude])
 wlr <- lm(formula = c ~ ., weights = nu, data = vars)
 rss <- sum(nu*residuals(wlr)^2 )
 J <- nrow(mf)
 A <- 2*(J - sum(1/trials))
 z <- (pearson - (J - ncol(vars) - 1))/sqrt(A + rss)
 p.value <- 2*(1 - pnorm(abs(z)))
 cat("z = ", z, "with p-value = ", p.value, "\n")
}

# Stukel Test
# Based on description in Hosmer and Lemeshow (2000) p. 155.
# Assumes data are aggregated into Explanatory Variable Pattern form.

stukel.test = function(obj) {
 # first, check to see if we fed in the right kind of object
 stopifnot(family(obj)$family == "binomial" && family(obj)$link == "logit")
 high.prob <- (obj$fitted.values >= 0.5)
 logit2 <- obj$linear.predictors^2
```

```
 z1 = 0.5*logit2*high.prob
 z2 = 0.5*logit2*(1-high.prob)
 mf <- obj$model
 trials = rep(1, times = nrow(mf))
 if(any(colnames(mf) == "(weights)"))
  trials <- mf[[ncol(mf)]]
 prop = mf[[1]]
 # the double bracket (above) gets the index of items within an object
 if (is.factor(prop))
  prop = (as.numeric(prop) == 2)  # Converts 1-2 factor levels to logical 0/1 values
 pi.hat = obj$fitted.values
 y <- trials*prop
 exclude <- which(colnames(mf) == "(weights)")
 vars <- data.frame(z1, z2, y, mf[,-c(1,exclude)])
 full <- glm(formula = y/trials ~ ., family = binomial(link = logit), weights = trials, data = vars)
 null <- glm(formula = y/trials ~ ., family = binomial(link = logit), weights = trials,
             data = vars[,-c(1,2)])
 LRT <- anova(null,full)
 p.value <- 1 - pchisq(LRT$Deviance[[2]], LRT$Df[[2]])
 cat("Stukel Test Stat = ", LRT$Deviance[[2]], "with p-value = ", p.value, "\n")
}


##########################################################################################################


##########################################################################
# NAME: Nikola Surjanovic and Peter Tea                                  #
# DATE: March 24, 2020                                                   #
# PURPOSE: C_n, and Information Matrix (IM) test statistics              #
#                                                                        #
# NOTES: Works for binary response regression models,                    #
#   with NO replications. Adapted from the "AllGOFTests.R" code          #
#   by Tom Loughin.                                                      #
##########################################################################

Cn.test <- function(obj) {
  # First, check to see if we fed in the right kind of object
  stopifnot(family(obj)$family == "binomial" && family(obj)$link == "logit")
  dat <- model.matrix(obj)
  y <- model.frame(obj)[ , 1]
  p <- ncol(dat)
  n <- nrow(dat)
  mu.hat <- obj$fitted.values
  nu.hat <- obj$linear.predictors

  C_n <- sum((y-mu.hat)/sqrt(mu.hat*(1-mu.hat)))
  Omega_n <- 0
  for (i in 1:n) {
    Omega_n <- Omega_n + exp(2*nu.hat[i])/((1+exp(nu.hat[i]))^4*mu.hat[i]*(1-mu.hat[i]))*
      (dat[i, ] %*% t(dat[i, ]))
  }
  Omega_n <- 1/n * Omega_n
```

```
  v_n <- 0
  for (i in 1:n) {
    v_n <- v_n + exp(nu.hat[i])/(1+exp(nu.hat[i]))^2/sqrt(mu.hat[i]*(1-mu.hat[i])) * dat[i, ]
  }
  v_n <- 1/n * v_n

  sigma2_n <- 1 - t(v_n) %*% Omega_n %*% v_n
  z <- C_n/sqrt(n*sigma2_n)
  P <- 2*pnorm(abs(z), lower.tail=FALSE)

  return(structure(list(
    method = c("C_n Statistic Goodness-of-Fit Test"),
    data.name = deparse(substitute(obj)),
    statistic = c(Z = z),
    p.value = P
  ), class = 'htest'))
}


IM.test <- function(obj) {
  # First, check to see if we fed in the right kind of object
  stopifnot(family(obj)$family == "binomial" && family(obj)$link == "logit")
  dat <- model.matrix(obj)
  y <- model.frame(obj)[ , 1]
  p <- ncol(dat)
  mu.hat <- obj$fitted.values

  chisq <- sum((y - mu.hat)*(1-2*mu.hat)*rowSums(dat^2))
  P <- pchisq(chisq, df=p, lower.tail=FALSE)

  return(structure(list(
    method = c("Information-Matrix (White 1982) Goodness-of-Fit Test"),
    data.name = deparse(substitute(obj)),
    statistic = c(X2 = chisq),
    parameter = c(df = p),
    p.value = P
  ), class = 'htest'))
}
```

Code for Plots in Introduction and Stukel Test Section

```
set.seed(320920)
n <- 40
x <- runif(n, -6, 6)
betas <- c(0, 0.7)
mu <- 1/(1+exp(-(betas[1] + betas[2]*x)))
y <- rbinom(n, size=1, prob=mu)

obj <- glm(y~x, family=binomial(link='logit'))


# For slides
plot(x=x,y=y, type='p', col='black', lwd=3, xlim=c(-6,6), ylab=expression(pi))
curve(1/(1+exp(-(betas[1] + betas[2]*x))), col='red', lwd=3, add=TRUE)

# For report
plot(x=x,y=y, type='p', col='black', lwd=5, xlim=c(-6,6), ylab=expression(pi), cex.axis=2, cex.lab=2)
curve(1/(1+exp(-(betas[1] + betas[2]*x))), col='red', lwd=5, add=TRUE)


# Recreate some plots from Stukel's paper

h <- function(x, alpha1, alpha2) {
  # x is a vector (potentially)!
  n <- length(x)
  ans <- numeric(n)
  for (i in 1:n) {
    eta <- x[i]
    if (eta>=0) {
      if (alpha1 > 0) {
        ansi <- 1/alpha1 * (exp(alpha1*abs(eta)) - 1)
      } else if (alpha1 == 0) {
        ansi <- eta
      } else {
        ansi <- -1/alpha1 * log(1-alpha1*abs(eta))
      }
    } else {
      if (alpha2 > 0) {
        ansi <- -1/alpha2 * (exp(alpha2*abs(eta)) - 1)
      } else if (alpha2 == 0) {
        ansi <- eta
      } else {
        ansi <- 1/alpha2 * log(1- alpha2*abs(eta))
      }
    }
    ans[i] <- ansi
  }
  return(ans)
}


# For slides
```

```r
curve(1*x, from=-3, to=3, xlab=expression(eta), ylab=expression(h(eta)), lwd=3)
curve(h(x, 0.25, 0.25), add=TRUE, lty=2, col='blue', lwd=3)
curve(h(x, -1, -1), add=TRUE, lty=3, col='red', lwd=5)


curve(exp(x)/(1+exp(x)), from=-6, to=6, xlab=expression(eta), ylab=expression(mu(eta)), lwd=3)
curve(1/(1+exp(-h(x, 0.25, 0.25))), add=TRUE, lty=2, col='blue', lwd=3)
curve(1/(1+exp(-h(x, -1, -1))), add=TRUE, lty=3, col='red', lwd=5)



# For report
par(mar=c(5.1, 5.1, 4.1, 2.1)) # Default: 5.1, 4.1, 4.1, 2.1
curve(1*x, from=-3, to=3, xlab=expression(eta), ylab=expression(h(eta)), lwd=5, cex.axis=2, cex.lab=2)
curve(h(x, 0.25, 0.25), add=TRUE, lty=2, col='blue', lwd=5)
curve(h(x, -1, -1), add=TRUE, lty=3, col='red', lwd=5)


curve(exp(x)/(1+exp(x)), from=-6, to=6, xlab=expression(eta), ylab=expression(mu(eta)), lwd=5,
      cex.axis=2, cex.lab=2)
curve(1/(1+exp(-h(x, 0.25, 0.25))), add=TRUE, lty=2, col='blue', lwd=5)
curve(1/(1+exp(-h(x, -1, -1))), add=TRUE, lty=3, col='red', lwd=5)

par(mar=c(5.1, 4.1, 4.1, 2.1))
```

Code for GOF Test Application Section

```
setwd("/Users/petertea/Documents/SFU/STAT 851/Project")
### Data Processing Steps
# --> Obtain: Service games of Djokovic in 2020 Aus Open.
# --> Variables of interest: Speed, Point won (yes/no), opponent name
library(dplyr)
library(ggplot2)


Player = "N. Djokovic"
year = 2020
tournament = "ausopen"


# -->  Read match data and points data, based on year and grand-slam tourney

sackmann_url = "https://raw.githubusercontent.com/JeffSackmann/tennis_slam_pointbypoint/master/"
matches_file_name <- paste(sackmann_url, as.character(year), "-", as.character(tournament),
                           "-matches.csv", sep=""  )
points_file_name <- paste(sackmann_url, as.character(year), "-", as.character(tournament),
                          "-points.csv", sep=""  )

#--> Select matches with the Player involved
points_data <- read.csv(points_file_name)
matches_data <- read.csv(matches_file_name)


which_match_id <- which(matches_data$player1 == Player | matches_data$player2 == Player)



match_ids <- matches_data[which_match_id,1]

Player_points_data <- points_data %>%
  dplyr::filter(match_id %in% match_ids)


# --> Identify whether the player of interest is coded as "player1" or "player2"
half_data <- matches_data %>%
  dplyr::filter(player1 == Player | player2 == Player) %>%
  dplyr::mutate(PointServer = ifelse(player1 == Player, 1 , 2),
                player_name = Player,
                opponent_name = player1) %>%
  dplyr::select(match_id, PointServer, player_name, opponent_name)



# --> Add column identifying which player (player1 or player2) is the PLAYER
Complete_Player_points_dat <- Player_points_data %>%
  dplyr::left_join( half_data,
                    by = c("match_id" = "match_id", "PointServer" = "PointServer")
  ) %>%
```

```
  dplyr::select(match_id,PointNumber, PointWinner,
                PointServer, Speed_KMH, P1Score, P2Score,
                RallyCount, ServeNumber, player_name, opponent_name)




Complete_Player_points_dat <- Complete_Player_points_dat[complete.cases(Complete_Player_points_dat),]
Complete_Player_points_dat$opponent_name <- as.character(Complete_Player_points_dat$opponent_name )
Complete_Player_points_dat$opponent_name <- as.factor(Complete_Player_points_dat$opponent_name )
levels(Complete_Player_points_dat$opponent_name) <- as.character(half_data$opponent_name)



#Calculate proportion of success (wins) in 1st service games vs 2nd service games
serve_success_rates <- function(mydata){
  library(dplyr)
  total <- mydata %>%
    group_by(ServeNumber) %>%
    tally()

  wins <- mydata %>%
    group_by(ServeNumber) %>%
    summarise(num_wins =sum(PointWinner==PointServer))

  result <- total %>%
    left_join(wins, by = "ServeNumber") %>%
    mutate(win_percentage = num_wins / n)

  colnames(result) = c("Serve Number", "Service Games", "Service Wins", "Win Percentage")

  return(result)
}

  ##############################################################################################

 ### Plot Serve Speed, and observed proportion of wins at that speed

Djoker_data <- Complete_Player_points_dat %>% dplyr::filter(Speed_KMH > 0)

Djoker_data <- Djoker_data %>%
    dplyr::mutate(Won = ifelse(PointServer == PointWinner, 1, 0))

Djoker_data %>%
  group_by(Speed_KMH, opponent_name) %>%
  summarise(Win_P = sum(PointWinner == PointServer)/n(),
            Count = n()) %>%
  ggplot(aes(x = Speed_KMH, y = Win_P, size = Count)) +

    #E56F0D: orange
    geom_point(color = "#0C5827", alpha = 0.7) +
    geom_point(shape = 1,colour = "black", alpha = 0.1)+
    ggtitle("Djokovic Australian Open 2020 \n Serve Speed Performance") +
```

```
    xlab("Speed (KM/H)") + ylab("Observed Win Proportion") +# labs(color = "Serve Number" ) +
    theme_classic() +
    #scale_fill_manual(values=c("#0C5827", "#fc9559"), labels = c("1st", "2nd") ) +
    #Adjust size of bubble points and legend title
    scale_size(breaks = 5, name="Number of \nReplicates") +
    theme(panel.background = element_rect(fill = "#f3f6fc", # background colour
                                          colour = "black", # border colour
                                          size = 0.5, linetype = "solid"),
          plot.title=element_text(size = rel(1.6),
                                  face = "bold", hjust = 0.5),
          legend.position = "right",
          legend.background = element_rect(colour = "#f3f6fc"),
          legend.key = element_rect(fill = "#f3f6fc"),
          axis.title = element_text(face = "bold", size = 13))#+ coord_fixed(ratio = 60)




# Some summary stats
length(unique(Djoker_data$Speed_KMH))
nrow(Djoker_data)




  ### Serve speed vs observed win proportion against all 7 opponents

  Djoker_data %>%
    group_by(Speed_KMH, opponent_name) %>%
    summarise(Win_P = sum(PointWinner == PointServer)/n(),
              Count = n()) %>%
    ggplot(aes(x = Speed_KMH, y = Win_P, size = Count)) +
    geom_point(color = "#0C5827", alpha = 0.75) +
    facet_wrap(~opponent_name, nrow = 3) +
    ggtitle("Djokovic AusOpen 2020 \n Performance Stratified by Opponent") +
    xlab("Speed (KM/H)") + ylab("Observed Win Proportion") +# labs(color = "Serve Number" ) +
    theme_classic() +
    scale_fill_manual(values=c("#0C5827", "#fc9559"), labels = c("1st", "2nd") ) +
    theme(panel.background = element_rect(fill = "#f3f6fc", # background colour
                                          colour = "black", # border colour
                                          size = 0.5, linetype = "solid"),
          plot.title=element_text(size = rel(1.6),
                                  face = "bold", hjust = 0.5),
          legend.position = "bottom",
          legend.background = element_rect(colour = "gray"),
          legend.key = element_rect(fill = "gray90"),
          axis.title = element_text(face = "bold", size = 13))



  ## Interaction effect between serve speed and opponent?

  #How fast did Djokovic serve against his 7 opponents,
```

```r
  # en route to the 2020 Australian Open Championship?
  #Did he make any speed adjustments against his opponents?


  # Plot serve speed distributions
  ggplot(Complete_Player_points_dat, aes(x=Speed_KMH)) +
    geom_density( aes(y = ..density..), fill ="#0C5827",  alpha = 0.6) +
    facet_wrap(~opponent_name, nrow = 2) +
    ggtitle("Djokovic AusOpen 2020 \n 1st Serve Speed vs. 2nd Serve Speed") +
    xlab("Speed (KM/H)") + ylab("Density") + #labs(fill = "Serve Number" ) +
    theme_classic() +
    #scale_fill_manual(values=c("#0C5827", "#fc9559"), labels = c("1st", "2nd") ) +
    theme(panel.background = element_rect(fill = "#f3f6fc", # background colour
                                          colour = "black", # border colour
                                          size = 0.5, linetype = "solid"),
          plot.title=element_text(size = rel(1.6),
                                  face = "bold", hjust = 0.5),
          legend.position = "bottom",
          legend.background = element_rect(colour = "gray"),
          legend.key = element_rect(fill = "gray90"),
          axis.title = element_text(face = "bold", size = 13))


  # Plot serve speed distributions
  ggplot(Complete_Player_points_dat, aes(x=Speed_KMH, fill=as.factor(ServeNumber) )) +
    geom_density( aes(y = ..density.., fill=as.factor(ServeNumber)), alpha = 0.6) +
    facet_wrap(~opponent_name, nrow = 2) +
    ggtitle("Djokovic AusOpen 2020 \n 1st Serve Speed vs. 2nd Serve Speed") +
    xlab("Speed (KM/H)") + ylab("Density") + labs(fill = "Serve Number" ) +
    theme_classic() +
    scale_fill_manual(values=c("#0C5827", "#fc9559"), labels = c("1st", "2nd") ) +
    theme(panel.background = element_rect(fill = "#f3f6fc", # background colour
                                          colour = "black", # border colour
                                          size = 0.5, linetype = "solid"),
          plot.title=element_text(size = rel(1.6),
                                  face = "bold", hjust = 0.5),
          legend.position = "bottom",
          legend.background = element_rect(colour = "gray"),
          legend.key = element_rect(fill = "gray90"),
          axis.title = element_text(face = "bold", size = 13))
```

```r
##############################################################################################
  # Fit model


  speed_model2 <- glm(data = Djoker_data,
                      formula = Won ~ Speed_KMH , family = binomial(link='logit'))
```

```
  temp2 = Djoker_data %>%
    group_by(Speed_KMH, opponent_name) %>%
    summarise(obs = sum(Won)/n()) #%>%
  #dplyr::mutate(fit = unique(fitted(speed_model2)))
  temp2$fit = unique(fitted(speed_model2))

  nrow(temp2)

  ggplot(temp2, aes(x = obs, y = fit )) +
    geom_point(color = "blue") +
    geom_smooth(method=lm, se=FALSE, formula=y~x-1, color = "black") +
    ggtitle("Observed Proportion vs. Fitted Probability") +
    xlab("Observed Proportion") + ylab("Fitted Probability") +
    theme_classic() +
    theme(panel.background = element_rect(fill = "#f3f6fc", # background colour
                                          colour = "black", # border colour
                                          size = 0.5, linetype = "solid"),
          plot.title=element_text(size = rel(1.6),
                                  face = "bold", hjust = 0.5),
          axis.title = element_text(face = "bold", size = 13))



#### GOF Testing
source("AllGOFTestsPlus.R")

HLTest(speed_model2, g = 10)
HLTest(speed_model2, g = 11)


o.r.test(speed_model2)
stukel.test(speed_model2)
Cn.test(speed_model2)
```