Biostatistics Lecture Notes

Generalized Linear Model

Ming-Chieh Shih

April 11, 2023

1 Starting from Linear Models

To get into generalized linear models, first we do a brief recap on linear models. Suppose we want to model the relationship between *one* outcome variable and *multiple* explanatory variables. We introduce the following notations:

- Y is the random variable for the outcome of interest
- $X = (1, X_1, X_2, ..., X_{p-1})$ is the random *row* vector of length p for the explanatory variables. We let the first variable be fixed as 1 to automatically include the intercept term.
- $\beta = (\beta_0, \beta_1, \beta_2, ..., \beta_{p-1})^\top$ is the regression coefficient vector, where β_0 is the intercept and β_k is the regression coefficient for X_k
- ε is the random variable for error

The basic and minimal assumption for (multiple) linear regression is then

$$Y = X\beta + \varepsilon \tag{1}$$
$$\mathbb{E}[\varepsilon|X] = 0 \tag{2}$$

We can see that conditional of X,

$$\mathbb{E}[Y|X] = \mathbb{E}[X\beta + \varepsilon|X] = X\beta + \mathbb{E}[\varepsilon|X] = X\beta \tag{3}$$
$$var[Y|X] = var[X\beta + \varepsilon|X] = var[\varepsilon|X] \tag{4}$$

Therefore, this formulation simply decomposes Y into the *mean component* $X\beta$ and the *random component* ε, and the only assumption made is *the mean of* Y *conditional on* X *is a linear combination of its elements*. Note that the mean of ε constrained to be 0 is to ensure the identifiability of β_0.

The most straightforward estimator for β is the ordinary least squares (OLS) estimator, which finds $\hat{\beta}$ that minimizes sum of squared residuals, $(Y - X\hat{\beta})^\top (Y - X\hat{\beta})$. Formally, suppose we draw **independent** samples from (Y, X) and yield the sample vector and matrix $(\boldsymbol{Y}, \boldsymbol{X})$, with $\boldsymbol{X}$ assumed to be rank p, then the OLS estimator is given by

$$\hat{\beta} = \arg\min_{\beta} (\boldsymbol{Y} - \boldsymbol{X}\beta)^\top (\boldsymbol{Y} - \boldsymbol{X}\beta) = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{Y} \tag{5}$$

The OLS estimator has a bunch of good properties, but note that some properties only hold under additional assumptions:

1. Unbiasedness: It can be easily shown that Equations (1) and (2) suffices in ensuring $\hat{\beta}$ is unbiased for β.
2. Best linear unbiased estimator (BLUE): The term "best" or "optimal" in statistics usually implies that the estimator has smaller variance than other estimators. To ensure that the OLS estimator is the best in some sense, we need to further assume that the error ε is *homoscedastic*. That is, the error variance of each observation should be identical. With this additional assumption, the *Gauss-Markov Theorem* ensures that $\hat{\beta}$ is the best linear unbiased estimator (BLUE), meaning that among all unbiased estimators that are linear combination

of Y, $\hat{\beta}$ has the smallest variance. Also, with $var[\varepsilon|X] = \sigma^2$ implied by homoscedasticity, we now have a concrete variance expression for $\hat{\beta}$:

$$var(\hat{\beta}) = (\boldsymbol{X}^\top \boldsymbol{X})^{-1}\sigma^2 \tag{6}$$

where σ^2 can be estimated empirically with the residuals.
An observation aside here is that when the error variance is *heteroscedastic*, i.e. each observation has its own error variance, then OLS is no longer the BLUE. Heuristically, this is because in OLS, when we are minimizing the sum of squared residuals, we are treating all residuals as equally important. However, if the error variance is heteroscedastic, some residuals have larger error variance and carry less information, so intuitively we should down-weight these terms when minimizing the squared residuals. Actually, it can be shown that if the error variance for the i^{th} observation is v_i, then we should use the *inverse error variance* as weights for the residuals, i.e. do the following optimization (writing $\boldsymbol{V} = diag(v_1, v_2, ..., v_n)$):

$$\arg\min_{\beta} \sum_i \frac{(\boldsymbol{Y}_i - \boldsymbol{X}_i\beta)^2}{v_i} = \arg\min_{\beta} (\boldsymbol{Y} - \boldsymbol{X}\beta)^\top \boldsymbol{V}^{-1}(\boldsymbol{Y} - \boldsymbol{X}\beta) = (\boldsymbol{X}^\top \boldsymbol{V}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^\top \boldsymbol{V}^{-1}\boldsymbol{Y} \tag{7}$$

This is called the *weighted least squares* (WLS) estimator, which can be shown to be the BLUE under heteroscedasticity. In generalized linear model, since the error variance is mostly heteroscedastic, WLS will come up once more but with a façade.

3. Uniform minimum-variance unbiased estimator (UMVUE): In BLUE, OLS is only gauranteed to be "better" than linear combination type estimators. However, if we are further willing to assume that ε is normally distributed so that $\varepsilon \sim N(0, \sigma^2)$, then the OLS estimator becomes the uniform minimum-variance unbiased estimator. That is, it is now the estimator with the smallest variance among *all* unbiased estimators. This fantastic property holds due to the fact that normal distribution belongs to the *exponential family*, and minimally sufficient statistics for exponential families are *complete*. Since the OLS estimator is a function of $\boldsymbol{X}^\top \boldsymbol{Y}$, a minimally sufficient statistic for β, and we have also shown that it is unbiased, *Lehmann-Scheffé Theorem* tells us that the OLS estimator is the UMVUE. In addition, assuming normality of ε enables us to not only know the variance of $\hat{\beta}$, but also the exact distribution of $\hat{\beta}$, so that we can construct exact tests and confidence intervals.

2 Maximum Likelihood Estimation for Linear Models

In the previous section, we constructed the OLS estimator from the intuition of minimizing the sum of squared residuals, after which we added the homoscedasticity and normality assumptions to ensure the OLS estimator is UMVUE. However, we can also work backwards and assume normality (and possibly homoscedasticity) first and try to come up with an estimator from it. In this case, we start from knowing the exact distribution of Y, so we have the luxury to work with likelihoods and derive maximum likelihood estimators (MLEs) to estimate β.

We first note that equivalently, we can also write Equations (1) and (2) as

$$Y_i|X_i \sim f(\theta_i) \tag{8}$$
$$\mathbb{E}[Y_i|X_i] := \mu_i = X_i\beta \tag{9}$$

so that f is an arbitrary distribution with parameters θ, with its mean μ assumed to be equal to $X\beta$. In the homoscedastic linear regression case, we let f be a normal distribution with parameters $\theta = (\mu, \sigma^2)^\top$ and σ^2 shared among observations, so that

$$Y_i|X_i \sim N(\mu_i, \sigma^2) \tag{10}$$
$$\mathbb{E}[Y_i|X_i] = \mu_i = X_i\beta \tag{11}$$

This is equivalent to just writing

$$Y_i|X_i \sim N(X_i\beta, \sigma^2) \tag{12}$$

Therefore, now we can write the log-likelihood for the observed data $(\boldsymbol{Y}, \boldsymbol{X})$, which is,

$$\ell(\beta, \sigma^2) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_i (\boldsymbol{Y}_i - \boldsymbol{X}_i\beta)^2 = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}(\boldsymbol{Y} - \boldsymbol{X}\beta)^\top(\boldsymbol{Y} - \boldsymbol{X}\beta) \tag{13}$$

We can obtain the maximum likelihood estimators for β and σ^2 by setting $\partial\ell/\partial\beta$ and $\partial\ell/\partial\sigma^2$ to zero (In theory we have to also look at second derivatives to ensure we're obtaining the maximum, but here we defer that). The former yields:

$$\left.\frac{\partial\ell}{\partial\beta}\right|_{\beta=\hat{\beta},\sigma^2=\hat{\sigma}^2} = -\frac{1}{\hat{\sigma}^2}(\boldsymbol{X}^\top\boldsymbol{X}\hat{\beta} - \boldsymbol{X}^\top\boldsymbol{Y}) = 0 \tag{14}$$

So we have the MLE for β, $\hat{\beta} = (\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top\boldsymbol{Y}$, which is exactly the same as the OLS estimator.

In later sections, we will talk about statistical properties of tests and estimates based on MLE. Unfortunately, most these properties are based on asymptotics, meaning that we only have a grasp of how MLE-related inference will behave in general when the sample size is large. Therefore, given the rich literature on the *exact* behavior of tests and estimates in linear models (a lot of then based on t and F distributions), the reformulation to maximum likelihood estimation above may seem unfruitful. However, our effort serves as a bridge to extend linear models to non-normal outcomes, under which MLE will be the preferred estimation methods and we will nearly always rely on asymptotics to do inference.

3 Problems of Linear Models under Non-normal outcomes

OLS estimators work great when the conditional distribution of the outcome, $Y|X$, can be modelled by normal distributions and the mean $\mathbb{E}[Y|X]$ can be modelled as a linear combination of X. In the case where there are non-linear relationships between elements of X and Y, we can try to pre-transform elements in X such as including square terms, cube terms, log terms, ... etc. before applying the OLS estimator. However, in biomedical studies, the nature of the outcome of interest can often violate these two assumption. For example, suppose we are interested in the occurrence or absence of an event, which makes Y a binary variable. We would encounter problems in two fronts:

1. **The problem in mean component**:
   Since Y is a binary variable, we have

   $$\mathbb{E}[Y|X] = \mathbb{P}[Y = 1|X] \tag{15}$$

   Therefore, $\mathbb{E}[Y|X]$ is a probability and should be in $[0, 1]$. However, in linear models we model $\mathbb{E}[Y|X]$ with $X\beta$, which can well be outside of $[0, 1]$. For example, if there is only one explanatory variable, age, and the estimated model is

   $$\hat{\mathbb{E}}[Y|X] = 0.05 + 0.015 \times \text{age} \tag{16}$$

   Then we would expect a disease probability of 1.75 for a subject aged 75 years old, which is improbable. This scenario would be even more prevalent when we include several explanatory variables. It would be desirable if the model could automatically eliminate this kind of scenario. A side point to note is that, although the scenario above is undesirable, under the case where the variation of event probability is not too large among observations, and the explanatory variables is also not heavy-tailed, modelling probabilities with vanilla linear combinations of X can still be reasonable. In addition, it has the advantage of easy interpretation of regression coefficients: an increase in X_k by one unit leads to a $\hat{\beta}_k$ increase in event probability. This is called a *linear probability model*, and is more often used among econometricians. All in all, we need to remember the famous quote: "*All models are wrong, but some are useful*".

2. **The problem in random component**:
   We mentioned that for the OLS estimator to be BLUE, the error ε should be *homoscedastic*, meaning that the error variance should be same among every observation. If ε is heteroscedasitc, then β would still be unbiased (given that $\mathbb{E}[Y|X]$ is adequately modelled), but the OLS estimator is no longer efficient. When $Y|X$ is not normally distributed, the error variance of $Y|X$ is often related to the mean $\mathbb{E}[Y|X]$, so the error is destined to be heteroscedastic unless every observation has the same mean. For example, when Y is binary, $Y|X$ may follow a Bernoulli distribution, so its variance is

   $$var[Y|X] = \mathbb{P}[Y = 1|X]\mathbb{P}[Y = 0|X] = (X\beta)(1 - X\beta) \tag{17}$$

which nearly guarantees heteroscedasticity. In this case, as we have mentioned around Equation (7), a WLS estimator using the inverse of variance as weights would perform better than the OLS estimator, and since the weights now depends on β, we would need an iterative procedure that alternates between estimating β with WLS and plugging in the estimated β into the weights. (In fact, it can be shown that combining this iterative procedure with non-linear mean component leads to one of the algorithms used in generalized linear models!)

With the two main problems in mind, we can now introduce the backbone structure of generalized linear models (GLMs), which served to tackle these two problems in one piece.

4 Model Structure of Generalized Linear Models

In the previous section, we put forward two problems in fitting non-normal outcomes with linear regression: (1) problems in the mean component, which stems from limited possible values for the outcome mean, and (2) problems in the random component, which stems from the heteroscedaticity naturally introduced by the distribution of the outcome. In generalized linear model, these two problems are addressed by specifying suitable outcome distributions and adequately transformed mean components. Namely, we have the general model structure:

$$Y_i|X_i \sim f(\theta_i) \tag{18}$$

$$g(\mu_i) = \eta_i := X_i\beta \tag{19}$$

where $\mu_i := \mathbb{E}[Y_i|X_i]$ is the mean outcome of observation i given X_i.

Compared with the linear model structure Equations (10) and (11), we can see that Equation (18) allows $Y_i|X_i$ to have a distribution other than normal distribution, which address the random component problem. In principle, $f(.)$ can be any distribution that fits $Y_i|X_i$, but to guarantee good statistical behavior of MLEs and ease analytical derivation, we often assume $f(.)$ belongs to a broad family of distributions called the *exponential family*. We will talk more about this in the next subsection.

Equation (19) maps μ_i to $X_i\beta$ with a function $g(.)$, so that they do not need to share the same support like linear models do. Here, $g(.)$ is chosen so that it is *strictly monotone*, *differentiable* and preferably *bijective* from the support of μ_i to $\mathbb{R}$. The strict monotonicity avoids identifiability problems of β and preserves the ranking of $X\beta$, i.e. letting observations with larger $X_i\beta$ also have larger mean of Y_i (or the other way around). The differentiability ensures that we can differentiate the log-likelihood function during during maximum likelihood estimation, and that the maximum occurs at where the partial derivative is zero. When there is bijectivity from the support of μ_i to $\mathbb{R}$, $g(.)$ is invertible and each value for $X_i\beta$ corresponds to a value for μ_i. We will talk about commonly used link function under different types of outcome in later subsections. Note that in later sections when applicable, we automatically assume that all distributions describing Y is actually referring to the conditional distribution $Y|X$, so we omit $|X$ for brevity.

4.1 Exponential Family Distributions

The definition of exponential family distributions is intuitive: suppose the probability distribution (mass) function of a random variable Y is $f(y;\theta)$, where θ is the parameter. An initial naïve construction of the logarithm of $f(y;\theta)$ assumes that it contains two additive parts:

$$\log f(y;\theta) = \_\_\_\_ + C(\theta) + D(y) \tag{20}$$

We then add some "interactions" between y and θ, and the most simple interaction would be a multiplication between functions of them:

$$\log f(y;\theta) = A(y)B(\theta) + C(\theta) + D(y) \tag{21}$$

or, equivalently,

$$f(y;\theta) = \exp[A(y)B(\theta) + C(\theta) + D(y)] \tag{22}$$

Any distribution with probability distribution (mass) function express-able as above belongs to the exponential family. In the case where θ is a vector of length p, then the definition becomes

$$f(y;\theta) = \exp\Big[\sum_{i=1}^{p} A_i(y)B_i(\theta) + C(\theta) + D(y)\Big] \tag{23}$$

This definition may seem very restricted at first glance, but surprisingly, a lot of common distributions actually belongs to the exponential family, to name a few:

- Normal
- Bernoulli
- Binomial (known number of exps)
- Poisson
- Negative binomial (known number of fails)
- Gamma
- Chi-square
- Exponential
- Log-normal
- Inverse Gaussian

Now let's take a closer look at Equation (21), the log-likelihood of one-parameter exponential family. This log-likelihood can be greatly simplified if we could reduce $A(y)B(\theta)$ into just $y\theta$. Fortunately, most of the time $A(.)$ and $B(.)$ are both invertible, so we can achieve this by slightly tweaking the parameter and the outcome.

For $B(\theta)$ to reduce to θ, we can just reparameterize the parameter so that $B(\theta)$ becomes the "new" θ. For example, in a Bernoulli distribution with probability parameter p, the log-likelihood can be written as

$$\begin{aligned}\log f(y;p) &= \log(p^y(1-p)^{1-y}) \\ &= y\log p + (1-y)\log(1-p) \\ &= y\log\frac{p}{1-p} + \log(1-p)\end{aligned} \tag{24}$$

therefore we have $B(p) = \log\frac{p}{1-p}$. Now we treat $\theta = \log\frac{p}{1-p}$ as our new parameter and the log-likelihood can be rewritten as

$$\log f(y;p) = y\theta - \log(e^\theta + 1) \tag{25}$$

Since setting $B(\theta)$ as the new parameter would simplify the log-likelihood and ease future calculation of moments, $B(\theta)$ is often termed as the *natural parameter* or *canonical parameter*.

For $A(y)$ to reduce to y, we can transform the outcome so that we are modelling $A(Y)$ instead of Y. The log-likelihood for $A(Y)$ would need to include an extra term due to variable transformation, but this term would only depend on y and can be absorbed into $D(y)$. This new log-likelihood would have $A(y)$ as y, and is said to be of *canonical form*.

In generalized linear model, we focus on the *one-parameter canonical exponential family with dispersion*, where the log probability density (mass) distribution can be expressed as

$$\log f(y;\theta,\phi) = \frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi) \tag{26}$$

Here ϕ is the dispersion parameter, a parameter that is not of our prime interest but affects the variance of the outcome. Sometime ϕ inevitably appears when we reduce a two-parameter exponential family into a one-parameter exponential family, eg. fixing the variance of a normal distribution or the shape of a gamma distribution. By convention, we assume that ϕ is known and does not estimate it, at least initially. For example, the log probability density function for normal distribution $N(\mu,\sigma^2)$ can be written as:

$$\begin{aligned}\log f(y;\mu,\sigma^2) &= -\frac{1}{2}\log(2\pi\sigma^2) - \frac{(y-\mu)^2}{2\sigma^2} \\ &= \frac{y\mu - \frac{\mu^2}{2}}{\sigma^2} + \Big(-\frac{1}{2}\log(2\pi\sigma^2) - \frac{y^2}{2\sigma^2}\Big)\end{aligned} \tag{27}$$

where $\phi := \sigma^2$ naturally appear in the denominator. Several common distributions that belong to the one-parameter canonical exponential family with dispersion are listed below:

The definition of exponential family distributions implies that the support of Y does not depend on the parameter θ. If otherwise, the probability density / mass function would have a term $I(y \in S(\theta))$ where $S(\theta)$ is a set dependent on θ, and this would not fit the definition of exponential families. With this property in mind, let us define two functions that are crucial in the inference of MLE:

- **Log likelihood function**: $\ell(\theta;y) := \log f(y;\theta)$
- **Score function**: $U(\theta;y) := \frac{\partial}{\partial\theta}\ell(\theta;y)$

	$N(\mu, \sigma^2)$	$Bern(p)$	$Bin(n,p)/n$	$Pois(\lambda)$	$Gamma(\alpha, \beta)$
θ	μ	$\log \frac{p}{1-p}$	$\log \frac{p}{1-p}$	$\log \lambda$	$-1/(\alpha\beta)$
ϕ	σ^2	1	1	1	$1/\alpha$
$b(\theta)$	$\theta^2/2$	$\log(e^\theta + 1)$	$n \log(e^\theta + 1)$	e^θ	$-\log(-\theta)$
$c(y, \phi)$	$-[\log(2\pi\phi) - y^2/\phi]/2$	1	$\log \binom{n}{y}$	$-\log y!$	$\log(y/\phi)/\phi - \log y - \log(\Gamma(1/\phi))$

Table 1: Characteristics of common one-parameter canonical exponential family with dispersion (letting $a(\phi) = \phi$). Gamma distribution is parameterized so that α is for shape and β is for scale.

Suppose we denote $\mathcal{X}_Y$ as the support for Y, then we can derive the mean of the score function:

$$\begin{aligned}
\mathbb{E}_{Y\sim f(\theta)}[U(\theta; Y)] &= \int_{\mathcal{X}_Y} \Big(\frac{\partial}{\partial\theta}\ell(\theta; y)\Big) f(y; \theta) dy && (28)\\
&= \int_{\mathcal{X}_Y} \Big(\frac{\partial}{\partial\theta} \log f(y; \theta)\Big) f(y; \theta) dy && (29)\\
&= \int_{\mathcal{X}_Y} \frac{\frac{\partial}{\partial\theta} f(y; \theta)}{f(y; \theta)} f(y; \theta) dy && (30)\\
&= \int_{\mathcal{X}_Y} \frac{\partial}{\partial\theta} f(y; \theta) dy && (31)\\
&= \frac{\partial}{\partial\theta} \int_{\mathcal{X}_Y} f(y; \theta) dy && (32)\\
&= \frac{\partial}{\partial\theta} 1 = 0 && (33)
\end{aligned}$$

where the exchange of partial differentiation and integration in Equation (32) is valid since the support of Y does not depend on θ. Note that here the θ used to evaluate $U(\theta; Y)$ is the same as the θ driving the distribution of Y from which expectation was taken. That is, say θ_0 is the true value for θ, then this result only guarantees that the expectation of $U(\theta_0; Y)$ is 0. Plug in other values for θ in U then the expectation will *not* be 0.

Based on the result above, we can derive another identity by differentiating $\mathbb{E}_{Y\sim f(\theta)}[U(\theta; Y)]$ again by θ:

$$\begin{aligned}
0 = \frac{\partial}{\partial\theta}\mathbb{E}_{Y\sim f(\theta)}[U(\theta; Y)] &= \frac{\partial}{\partial\theta}\Big[\int_{\mathcal{X}_Y} \Big(\frac{\partial}{\partial\theta}\ell(\theta; y)\Big) f(y; \theta) dy\Big] && (34)\\
&= \int_{\mathcal{X}_Y} \frac{\partial}{\partial\theta}\Big[\Big(\frac{\partial}{\partial\theta}\ell(\theta; y)\Big) f(y; \theta)\Big] dy && (35)\\
&= \int_{\mathcal{X}_Y} \Big(\frac{\partial^2}{\partial\theta^2}\ell(\theta; y)\Big) f(y; \theta) + \Big(\frac{\partial}{\partial\theta}\ell(\theta; y)\Big)\Big(\frac{\partial}{\partial\theta} f(y; \theta)\Big) dy && (36)\\
&= \int_{\mathcal{X}_Y} \Big(\frac{\partial^2}{\partial\theta^2}\ell(\theta; y)\Big) f(y; \theta) dy + \int_{\mathcal{X}_Y} \Big(\frac{\partial}{\partial\theta}\ell(\theta; y)\Big) \frac{\frac{\partial}{\partial\theta} f(y; \theta)}{f(y; \theta)} f(y; \theta) dy && (37)\\
&= \int_{\mathcal{X}_Y} \Big(\frac{\partial^2}{\partial\theta^2}\ell(\theta; y)\Big) f(y; \theta) dy + \int_{\mathcal{X}_Y} \Big(\frac{\partial}{\partial\theta}\ell(\theta; y)\Big)^2 f(y; \theta) dy && (38)\\
&= \mathbb{E}_{Y\sim f(\theta)}\Big[\frac{\partial^2}{\partial\theta^2}\ell(\theta; Y)\Big] + \mathbb{E}_{Y\sim f(\theta)}[U^2(\theta; Y)] && (39)
\end{aligned}$$

Here the exchange of partial differentiation and integration sign is used again in Equation (35). Based on this result, we have the variance of the score function:

$$\begin{aligned}
var_{Y\sim f(\theta)}[U(\theta; Y)] &= \mathbb{E}_{Y\sim f(\theta)}[U^2(\theta; Y)] - (\mathbb{E}_{Y\sim f(\theta)}[U(\theta; Y)])^2 && (40)\\
&= \mathbb{E}_{Y\sim f(\theta)}[U^2(\theta; Y)] = -\mathbb{E}_{Y\sim f(\theta)}\Big[\frac{\partial^2}{\partial\theta^2}\ell(\theta; Y)\Big] && (41)
\end{aligned}$$

where Equation (40) to Equation (41) is based on Equation (33), and the equal sign in Equation (41) is based on Equation (39). The negative expecation of second derivative of log-likelihood function is an extremely imported quantity in likelihood inference, which is termed as the **Fisher information** notated as $\mathcal{I}(\theta)$.

Since for our family of distributions:

$$\frac{\partial \ell(\theta; Y)}{\partial \theta} = \frac{Y - b'(\theta)}{a(\phi)} \tag{42}$$

$$\frac{\partial^2 \ell(\theta; Y)}{\partial \theta^2} = -\frac{b''(\theta)}{a(\phi)} \tag{43}$$

Using Equation (33) we have:

$$0 = \mathbb{E}_{Y \sim f(\theta)}\Big[\frac{\partial \ell(\theta; Y)}{\partial \theta}\Big] = \frac{\mathbb{E}(Y) - b'(\theta)}{a(\phi)} \tag{44}$$

$$\mathbb{E}(Y) := \mu = b'(\theta) \tag{45}$$

And using Equation (41) we have:

$$var_{Y \sim f(\theta)}\Big[\frac{\partial}{\partial \theta}\ell(\theta; Y)\Big] = -\mathbb{E}_{Y \sim f(\theta)}\Big[\frac{\partial^2}{\partial \theta^2}\ell(\theta; Y)\Big] \tag{46}$$

$$\frac{var(Y)}{a^2(\phi)} = -\Big(-\frac{b''(\theta)}{a(\phi)}\Big) \tag{47}$$

$$var(Y) = b''(\theta)a(\phi) \tag{48}$$

Apart from verifying that Equations (45) and (48) are true using Table 1, we can now see why it makes sense to call ϕ the *dispersion parameter*: ϕ does not influence the conditional mean of Y, but only inflates (or shrinks) the conditional variance of Y. This fact will be important we later talk about overdispersion, a measure for goodness (or in fact "badness") of fit of our GLM.

4.2 Link Functions and Canonical Links

As we have elaborated, the original purpose of the link function $g(.)$ is to map μ_i to $X_i\beta$, which has different range of possible values. Under this sole purpose, there are actually multiple (if not infinitely many) link functions we can choose from, as long as it is strictly monotone, differentiable and does the mapping we desire. The choice between these link functions are actually mostly out of customs in the field, and one may also use model fit indices to select between link functions.

Another purpose of link functions is to simplify the model to make likelihood maximization easier and more reliable. Link functions of this kind are called *canonical links*, which is determined by letting the natural parameter $\theta = X\beta$. In this case, the log-likelihood becomes,

$$\log f(y; x, \beta, \phi) = \frac{yx\beta - b(x\beta)}{a(\phi)} + c(y, \phi) \tag{49}$$

which has a nice $yx\beta$ term so the derivative of log-likelihood with respect to β for the first term does not involve chain rule. We will talk about this more in the next section on estimation. Since with the canonical link $g(\mu(\theta)) = X\beta = \theta$, we yield that the canonical link is given by inverting the function of μ on θ, i.e. $g(.) = \mu^{-1}(.)$. A word of caution is that, canonical links may not guarantee bijection between the support of μ to the real line, as we will see in Gamma distribution. In the following, we will list some of the most commonly used links, categorized by the outcome distribution:

- **Normal distribution**:
  - Identity link ($g(\mu) = \mu$): This is also the canonical link.
- **Bernoulli distribution**:
  - Logit link ($g(\mu) = \log \frac{\mu}{1-\mu}$): This link has the advantage of easy-to-interpret regression coefficients as log odds ratios (more about that in upcoming lectures), and is therefore prevalent in the biomedical and machine learning field alike, i.e. logistic regression. It is also the canonical link.
  - Probit link ($g(\mu) = \Phi^{-1}(\mu)$): This link uses the inverse of normal cumulative distribution function as link function. Probit link is more popular in social sciences and econometrics since it corresponds to assuming each subject has a score based on their X, and whenever their score adding a normal noise exceeds a threshold, the subject will experience an event.

- Complementary log-log link ($g(\mu) = \log(-\log(1-\mu))$): This link is more prevalent in industrial statistics.

- **Poisson distribution**:
  - Log link ($g(\mu) = \log \mu$): This link is intuitive since it maps μ, a positive real number to $(-\infty, \infty)$. Under this link, the regression coefficients can be interpreted as log rate ratios (more about this in upcoming lectures). It is also the canonical link.

- **Gamma distribution**:
  - Log link ($g(\mu) = \log \mu$): Since the mean of Gamma distribution is also a positive number, it makes sense to use the log link. However, this is *NOT* the canonical link for Gamma distribution. Nevertheless, log link is still often used for its coefficient interpretability (log-ratio of mean outcome).
  - Negative reciprocal link ($g(\mu) = -1/\mu$): This is Gamma distribution's canonical link, which can be verified by seeing in Gamma distribution, $\mu = -1/\theta$. However, this link does not map positive numbers to the real line, so we would predict subjects with positive $X\beta$ to have negative mean outcomes, which is sometimes undesirable. It is harder to interpret the regression coefficients under this link.

5 Estimation of Generalized Linear Models

Here we first reiterate the model structure of generalized linear models (GLMs):

$$Y_i | X_i \sim f(\theta_i, \phi) \tag{50}$$
$$g(\mu_i) := \eta_i = X_i \beta \tag{51}$$

where $f(\theta_i, \phi)$ typically belongs to the *one-parameter canonical exponential family with dispersion*. Given that θ_i is the natural parameter, the log-likelihood implied by Equation (52) is

$$\log f(\boldsymbol{Y}_i | \theta_i, \phi) = \frac{\boldsymbol{Y}_i \theta_i - b(\theta_i)}{a(\phi)} + c(\boldsymbol{Y}_i, \phi) \tag{52}$$

Also, since μ_i is the conditional expectation of Y_i given X_i, we have derived the following relation:

$$\mu_i = b'(\theta_i) \tag{53}$$

Combining Equations (51) to (53), we have the explicit form of the log-likelihood function contributed by observation i:

$$\begin{aligned} \log f(\boldsymbol{Y}_i | \theta_i, \phi) &= \frac{\boldsymbol{Y}_i \theta_i - b(\theta_i)}{a(\phi)} + c(y_i, \phi) \\ \mu_i &= b'(\theta_i) \\ g(\mu_i) := \eta_i &= \boldsymbol{X}_i \beta \end{aligned} \tag{54}$$

5.1 The score function and generalized estimating equations

As we have elaborated in previous sections, the regression parameters of GLMs (β) are typically estimated via maximum likelihood estimation (MLE). More often than not, the MLE will occur at where the first derivative of the log-likelihood, i.e. score function is zero. A word of caution is that, previously when we defined the score function, we differentiated the log-likelihood with

respect to θ. But now we are interested in estimating β, so we will have a *different* score function:

$$U(\beta|\boldsymbol{Y}_i, \boldsymbol{X}_i, \phi) = \frac{\partial}{\partial \beta}\ell(\beta|\boldsymbol{Y}_i, \boldsymbol{X}_i, \phi) \tag{55}$$

$$= \frac{\partial \log f(\boldsymbol{Y}_i|\theta_i, \phi)}{\partial \theta_i}\frac{\partial \theta_i}{\partial \mu_i}\frac{\partial \mu_i}{\partial \eta_i}\frac{\partial \eta_i}{\partial \beta} \tag{56}$$

$$= \Big(\frac{\boldsymbol{Y}_i - b'(\theta_i)}{a(\phi)}\Big)\Big(\frac{\partial \mu_i}{\partial \theta_i}\Big)^{-1}\frac{\partial \mu_i}{\partial \eta_i}\boldsymbol{X}_i^\top \tag{57}$$

$$= \Big(\frac{\boldsymbol{Y}_i - b'(\theta_i)}{b''(\theta_i)a(\phi)}\Big)\frac{\partial \mu_i}{\partial \eta_i}\boldsymbol{X}_i^\top \tag{58}$$

$$= \underbrace{\Big(\frac{\boldsymbol{Y}_i - \mathbb{E}[Y_i|\boldsymbol{X}_i]}{var[Y_i|\boldsymbol{X}_i]}\Big)}_{\text{Weighted residual}} \quad \underbrace{\frac{\partial \mu_i}{\partial \eta_i}}_{\text{Link function correction}} \quad \underbrace{\boldsymbol{X}_i^\top}_{\text{Covariates}} \tag{59}$$

where $\frac{\partial \mu_i}{\partial \eta_i} = \big(\frac{\partial \eta_i}{\partial \mu_i}\big)^{-1} = \big(g'(\mu_i)\big)^{-1} = \big(g'(g^{-1}(X_i\beta))\big)^{-1}$.

The form of Equation (59) gives us two intuitions. First, previously we talked about weighted least squares (WLS), where if $v_i = var[Y_i|\boldsymbol{X}_i]$, we try to minimize the weighted sum of squared residuals using inverse variance as weights:

$$\sum_i \frac{(Y_i - \boldsymbol{X}_i\beta)^2}{v_i} \tag{60}$$

differentiating with respect to β yields

$$2\sum_i \Big(\frac{Y_i - \boldsymbol{X}_i\beta}{v_i}\Big)\boldsymbol{X}_i^\top \tag{61}$$

since in WLS we have $\mathbb{E}[Y_i|\boldsymbol{X}_i] = \boldsymbol{X}_i\beta$, Equation (61) is basically equivalent to the score function of GLMs with identity link (so that $\partial\mu_i/\partial\eta_i = 1$). In fact, as we will see, the most common estimation algorithm for GLMs, iterative weighted least squares (IWLS), is exactly developed using the estimator for WLS.

Second, notice that in Equation (59), as long as we have the form of conditional expectation and variance of Y_i, along with the link function and covariates, we can construct the score without even knowing what the likelihood looks like. In particular, we can matricize some of the notations as

$$\boldsymbol{V} = diag(var[Y_i|\boldsymbol{X}_i]) \tag{62}$$

$$\boldsymbol{D} = diag\Big(\frac{\partial \mu_i}{\partial \eta_i}\Big)\boldsymbol{X} \tag{63}$$

Then we can show that

$$\sum_i U(\beta|\boldsymbol{Y}_i, \boldsymbol{X}_i, \phi) = \boldsymbol{D}^\top\boldsymbol{V}^{-1}(\boldsymbol{Y} - \mathbb{E}[\boldsymbol{Y}|\boldsymbol{X}]) \tag{64}$$

This actually serves as a starting point for *generalized estimating equations (GEE)*, a model that allows observation in GLMs to be correlated. To introduce correlation between observations, instead of letting $\boldsymbol{V}$ be diagonal, we may replace $\boldsymbol{V}$ with

$$\boldsymbol{V}^* = \boldsymbol{V}^{1/2}\boldsymbol{R}\boldsymbol{V}^{1/2} \tag{65}$$

where $\boldsymbol{R}$ is the correlation matrix. The resulting score function, which is also termed as *quasi-score function*, is then solved for roots for $\hat{\beta}$. Since GEE is not an intended topic for this course, we will stop here and those interested can look for relative materials in the textbook (Ch 11) or online.

5.2 Newton-Raphson algorithm and Fisher scoring

Our goal now is to look for the solution for the equation

$$\sum_i U(\beta|\boldsymbol{Y}_i, \boldsymbol{X}_i, \phi) := \sum_i U_i(\beta) \tag{66}$$

$$= \sum_i \Big(\frac{\boldsymbol{Y}_i - b'(\theta_i)}{a(\phi)}\Big)\frac{\partial \theta_i}{\partial \mu_i}\frac{\partial \mu_i}{\partial \eta_i}\boldsymbol{X}_i^\top \tag{67}$$

$$= \sum_i \Big(\frac{\boldsymbol{Y}_i - b'(\theta_i)}{b''(\theta_i)a(\phi)}\Big)\frac{\partial \mu_i}{\partial \eta_i}\boldsymbol{X}_i^\top = 0 \tag{68}$$

Here we will use the most classic root-finding algorithm, the *Newton-Raphson* algorithm. Briefly, in Newton-Raphson algorithm, suppose we would like to find the root for a function $h(\beta)$, then we first give an initial ("0-th") guess for β, which we denote as $\hat{\beta}^{(0)}$. Then we iteratively guess for β using the following iterative equation

$$\hat{\beta}^{(k+1)} = \hat{\beta}^{(k)} - (h'(\hat{\beta}^{(k)}))^{-1} h(\hat{\beta}^{(k)}) \tag{69}$$

In our case, our $h(\beta)$ is $\sum_i U_i(\beta)$, so we have the relation

$$\hat{\beta}^{(k+1)} = \hat{\beta}^{(k)} + \Big(\sum_i - \frac{\partial U_i(\beta)}{\partial \beta^\top}\Big|_{\beta=\hat{\beta}^{(k)}} \Big)^{-1} \sum_i U_i(\hat{\beta}^{(k)}) \tag{70}$$

Notice that $-\frac{\partial}{\partial \beta^\top} U_i(\beta)$ is the negative second derivative of the log-likelihood with respect to β. Its expectation, as we have defined, is called the Fisher information $\mathcal{I}_i(\beta)$. Here we are not calculating its expectation rather than evaluating it at one instance $\boldsymbol{Y}_i$, so we term it as the *observed information*, denoted as $I_i(\beta)$.

Based on Equation (70), to use Newton-Raphson algorithm, for each iteration we would need to calculate *and* invert the sum of observed information, i.e.

$$\sum_i I_i(\hat{\beta}^{(k)}) = \sum_i - \frac{\partial U_i(\beta)}{\partial \beta^\top}\Big|_{\beta=\hat{\beta}^{(k)}} = \sum_i - \frac{\partial}{\partial \beta^\top} \Big[\Big(\frac{\boldsymbol{Y}_i - b'(\theta_i)}{b''(\theta_i)a(\phi)} \Big) \frac{\partial \mu_i}{\partial \eta_i} \Big] \Big|_{\beta=\hat{\beta}^{(k)}} \boldsymbol{X}_i^\top \tag{71}$$

Analytically, this is not an easy task since both θ_i and $\partial \mu_i / \partial \eta_i$ involves β, and differentiation with respect to β would be very complicated. Therefore, to use the vanilla form of Newton-Raphson algorithm, the derivative matrix may need to be solved numerically then inverted, which is very resource-demanding, at least for computers back in the 1980s.

To ease the computational burden, statisticians resorted to replacing the observed information $I_i(.)$ with its expectation, i.e. the Fisher information $\mathcal{I}_i(.)$. So now our iterative equation becomes

$$\hat{\beta}^{(k+1)} = \hat{\beta}^{(k)} + \Big(\sum_i \mathcal{I}_i(\hat{\beta}^{(k)}) \Big)^{-1} \sum_i U_i(\hat{\beta}^{(k)}) \tag{72}$$

Recall that previously we derived the following relation for distributions that has support independent to parameter θ:

$$\mathbb{E}_{Y \sim f(\theta)}[U^2(\theta)] = \mathcal{I}(\theta) \tag{73}$$

This relation was derived for a scalar parameter θ, but we can also derive it with respect to a vector parameter β and yield:

$$\mathbb{E}_{Y \sim f_i(\beta)}[U_i(\beta) U_i^\top(\beta)] = \mathcal{I}_i(\beta) \tag{74}$$

Therefore, we have (assuming $\beta = \hat{\beta}^{(k)}$, and taking the expectation under $Y_i \sim f_i(\hat{\beta}^{(k)})$)

$$\begin{aligned}
\mathcal{I}_i(\hat{\beta}^{(k)}) &= \mathbb{E}[U_i(\hat{\beta}^{(k)}) U_i^\top(\hat{\beta}^{(k)}) \Big| \boldsymbol{X}_i] && (75)\\
&= \mathbb{E}\Big[\Big(\frac{(\boldsymbol{Y}_i - b'(\theta_i))^2}{(b''(\theta_i)a(\phi))^2} \big(\frac{\partial \mu_i}{\partial \eta_i} \big)^2 \Big) \boldsymbol{X}_i^\top \boldsymbol{X}_i \Big| \boldsymbol{X}_i \Big] && (76)\\
&= \mathbb{E}\Big[\Big(\frac{(\boldsymbol{Y}_i - \mathbb{E}[\boldsymbol{Y}_i|\boldsymbol{X}_i])^2}{(var[\boldsymbol{Y}_i|\boldsymbol{X}_i])^2} \big(\frac{\partial \mu_i}{\partial \eta_i} \big)^2 \Big) \boldsymbol{X}_i^\top \boldsymbol{X}_i \Big| \boldsymbol{X}_i \Big] && (77)\\
&= \frac{var[\boldsymbol{Y}_i|\boldsymbol{X}_i]}{(var[\boldsymbol{Y}_i|\boldsymbol{X}_i])^2} \big(\frac{\partial \mu_i}{\partial \eta_i} \big)^2 \boldsymbol{X}_i^\top \boldsymbol{X}_i && (78)\\
&= \frac{1}{var[\boldsymbol{Y}_i|\boldsymbol{X}_i]} \big(\frac{\partial \mu_i}{\partial \eta_i} \big)^2 \boldsymbol{X}_i^\top \boldsymbol{X}_i && (79)\\
&= \frac{\boldsymbol{X}_i^\top \boldsymbol{X}_i}{b''(\theta_i)a(\phi)} \big(\frac{\partial \mu_i}{\partial \eta_i} \big)^2 && (80)\\
&= \frac{\boldsymbol{X}_i^\top \boldsymbol{X}_i}{b''(\mu^{-1}(g^{-1}(\boldsymbol{X}_i \hat{\beta}^{(k)})))a(\phi)} (g^{-1'}(\boldsymbol{X}_i \hat{\beta}^{(k)}))^2 && (81)
\end{aligned}$$

where $\mu(.)$ is the function relating $\mu_i = \mu(\theta_i)$. This expression is considerably easier to evaluate than the observed information matrix. This trick of replacing the second derivative with its expectation is termed as *Fisher scoring*, and is considered the norm of GLM estimation.

Recall that in the previous section, we defined the *canonical link* of an exponential family as the link that lets the natural parameter $\theta_i = X_i\beta$, which implies $g(.) = \mu^{-1}(.)$, or $g^{-1}(.) = \mu(.)$. Here we can finally demonstrate two advantages of using the canonical link. First, Equation (81) can be further simplified as

$$\mathcal{I}_i(\hat{\beta}^{(k)}) = \frac{\boldsymbol{X}_i^\top \boldsymbol{X}_i}{b''(\boldsymbol{X}_i\hat{\beta}^{(k)})a(\phi)}(\mu'(\boldsymbol{X}_i\hat{\beta}^{(k)}))^2 \tag{82}$$

$$= \frac{\boldsymbol{X}_i^\top \boldsymbol{X}_i}{b''(\boldsymbol{X}_i\hat{\beta}^{(k)})a(\phi)}(b''(\boldsymbol{X}_i\hat{\beta}^{(k)}))^2 \tag{83}$$

$$= \frac{b''(\boldsymbol{X}_i\hat{\beta}^{(k)})}{a(\phi)}\boldsymbol{X}_i^\top \boldsymbol{X}_i \tag{84}$$

which is extra nice. Second, since $\theta_i = X_i\beta := \eta_i$, we have $\frac{\partial\theta_i}{\partial\eta_i} = 1$, so the score function can now be simplified as, from Equation (67):

$$U_i(\beta) = \Big(\frac{\boldsymbol{Y}_i - b'(\theta_i)}{a(\phi)}\Big)\frac{\partial\theta_i}{\partial\mu_i}\frac{\partial\mu_i}{\partial\eta_i}\boldsymbol{X}_i^\top = \Big(\frac{\boldsymbol{Y}_i - b'(\theta_i)}{a(\phi)}\Big)\boldsymbol{X}_i^\top \tag{85}$$

Notice that since ϕ is assumed known and unrelated to β, the derivative of $U_i(\beta)$ does not depend on $\boldsymbol{Y}_i$ anymore. Therefore, the observed information would be identical to the Fisher information. To write it out explicitly:

$$I_i(\hat{\beta}^{(k)}) = -\frac{\partial U_i(\beta)}{\partial\beta^\top}\Big|_{\beta=\hat{\beta}^{(k)}} = -\frac{\partial}{\partial\beta^\top}\Big(\frac{\boldsymbol{Y}_i - b'(\theta_i)}{a(\phi)}\Big)\boldsymbol{X}_i^\top\Big|_{\beta=\hat{\beta}^{(k)}} \tag{86}$$

$$= \frac{\partial}{\partial\beta^\top}\frac{b'(\theta_i)}{a(\phi)}\boldsymbol{X}_i^\top\Big|_{\beta=\hat{\beta}^{(k)}} \tag{87}$$

$$= \frac{\partial}{\partial\beta^\top}\frac{b'(\boldsymbol{X}_i\beta)}{a(\phi)}\boldsymbol{X}_i^\top\Big|_{\beta=\hat{\beta}^{(k)}} \tag{88}$$

$$= \frac{b''(\boldsymbol{X}_i\beta)}{a(\phi)}\boldsymbol{X}_i^\top\boldsymbol{X}_i\Big|_{\beta=\hat{\beta}^{(k)}} \tag{89}$$

$$= \frac{b''(\boldsymbol{X}_i\hat{\beta}^{(k)})}{a(\phi)}\boldsymbol{X}_i^\top\boldsymbol{X}_i = \mathcal{I}_i(\hat{\beta}^{(k)}) \tag{90}$$

Therefore, when using canonical links, the vanilla Newton-Raphson algorithm is identical to Fisher scoring. However, this would not be so if non-canoncial links are used.

5.3 Iterative weighted least squares (Optional)

Let us get back to the Fisher scoring iterative equation, Equation (72). Notice that the sum of individual Fisher information can be matricized as follows (we assume canonical links are used for brevity),

$$\sum_i \mathcal{I}_i(\hat{\beta}^{(k)}) = \sum_i \frac{b''(\boldsymbol{X}_i\hat{\beta}^{(k)})}{a(\phi)}\boldsymbol{X}_i^\top\boldsymbol{X}_i = \boldsymbol{X}^\top\boldsymbol{W}(\hat{\beta}^{(k)})\boldsymbol{X} \tag{91}$$

$$\boldsymbol{W}(\beta) = diag\Big(\frac{b''(\boldsymbol{X}_i\beta)}{a(\phi)}\Big) \tag{92}$$

The sum of score functions can also be matricized as follows,

$$\sum_i U_i(\hat{\beta}^{(k)}) = \sum_i \Big(\frac{\boldsymbol{Y}_i - b'(\theta_i)}{a(\phi)}\Big)\boldsymbol{X}_i^\top = \boldsymbol{X}^\top\boldsymbol{W}(\hat{\beta}^{(k)})\boldsymbol{t}(\hat{\beta}^{(k)}) \tag{93}$$

$$\boldsymbol{t}(\beta) = diag\Big(\frac{1}{b''(\boldsymbol{X}_i\beta)}\Big)(\boldsymbol{Y} - b'(\boldsymbol{X}\beta)) \tag{94}$$

Combining these matrix notations, we can rewrite Equation (72) as

$$\boldsymbol{X}^\top\boldsymbol{W}(\hat{\beta}^{(k)})\boldsymbol{X}\hat{\beta}^{(k+1)} = \boldsymbol{X}^\top\boldsymbol{W}(\hat{\beta}^{(k)})\boldsymbol{X}\hat{\beta}^{(k)} + \boldsymbol{X}^\top\boldsymbol{W}(\hat{\beta}^{(k)})\boldsymbol{t}(\hat{\beta}^{(k)}) \tag{95}$$

Writing $\boldsymbol{z}(\hat{\beta}^{(k)}) = \boldsymbol{X}\hat{\beta}^{(k)} + \boldsymbol{t}(\hat{\beta}^{(k)})$, we then have

$$\boldsymbol{X}^\top\boldsymbol{W}(\hat{\beta}^{(k)})\boldsymbol{X}\hat{\beta}^{(k+1)} = \boldsymbol{X}^\top\boldsymbol{W}(\hat{\beta}^{(k)})\boldsymbol{z}(\hat{\beta}^{(k)}) \tag{96}$$

with the solution

$$\hat{\beta}^{(k+1)} = (\boldsymbol{X}^\top \boldsymbol{W}(\hat{\beta}^{(k)})\boldsymbol{X})^{-1}\boldsymbol{X}^\top \boldsymbol{W}(\hat{\beta}^{(k)})\boldsymbol{z}(\hat{\beta}^{(k)}) \tag{97}$$

Notice that this form is very similar to the solution of WLS shown in Section 1, where $\boldsymbol{z}$ is the outcome vector and $\boldsymbol{W}$ is the weight matrix. Since the weight matrix is changing for every iteration due to different $\hat{\beta}^{(k)}$, this method is termed as the *iterative weighted least squares*. This method was proposed to take advantage of the already well-developed WLS procedure in early years, so that statistician would not have to rewrite all the optimization details for GLMs.

6 Inference for Generalized Linear Models

Now we know how to estimate GLMs with MLE, the next question is then how do we do inference on GLMs? The inference we are interested in is basically the same as linear models (1) How do we test if any linear combination of regression coefficients β is equal to a given value (mostly 0)? How do we construct confidence intervals for them? (2) How do we compare the fit of two models? (3) How do we determine if the model is well-fitted or not? In the following, we will tackle these three queries one by one.

6.1 Asymptotic distribution for estimated regression coefficients

In linear regression, we are able to derive the *exact* distribution of regression coefficient estimators using t-distributions. However, in GLMs, we do not have the luxury of exact distributions. Instead, we have to look back to theories on MLEs and resort to asymptotics, i.e. looking into the case where the sample size is large. A good property of MLEs that can help us a ton is that under certain regulatory conditions (which GLMs often satisfy), MLEs are *consistent* estimators. That is, suppose β_0 is the true value of the regression coefficients, and let $\hat{\beta}_n$ be the MLE estimator of β based on sample size n, then for all entries of the vector, we have

$$\lim_{n\to\infty} \mathbb{P}(|\hat{\beta}_{ni} - \beta_{0i}| > \varepsilon) = 0, \quad \forall \varepsilon > 0 \tag{98}$$

Roughly speaking, as the sample size n grows large, there will be high probability that $\hat{\beta}_n$ will be within a tight vicinity of β. This in turn enables us to use Taylor approximations on functions of β to extract important inference detail. We first start with the score function $U(\beta) = \sum_i U_i(\beta)$, where we try to approximate $U(\beta_0)$. Since the MLE estimator $\hat{\beta}$ is consistent, asymptotically we can use first-order approximation to approximate $U(\beta_0)$ as follows:

$$\begin{aligned} U(\beta_0) &\approx U(\hat{\beta}) + U'(\hat{\beta})(\beta_0 - \hat{\beta}) && (99)\\ &= U(\hat{\beta}) - I(\hat{\beta})(\beta_0 - \hat{\beta}) && (100)\\ &\approx U(\hat{\beta}) - \mathcal{I}(\hat{\beta})(\beta_0 - \hat{\beta}) && (101) \end{aligned}$$

where in Equation (100) we used the definition of observed information as the negative second derivative of log-likelihood function, and in Equation (101) we approximate the observed information with its expectation, i.e. Fisher information. Now, using the fact that the $\hat{\beta}$ is found by letting $U(\hat{\beta}) = 0$, we have

$$\hat{\beta} - \beta_0 \approx \mathcal{I}^{-1}(\hat{\beta})U(\beta_0) \tag{102}$$

Recall that we have shown

$$\begin{aligned} \mathbb{E}[U_i(\beta_0)] &= 0 && (103)\\ var[U_i(\beta_0)] &= \mathcal{I}_i(\beta_0) && (104) \end{aligned}$$

Since each U_i comes from independently distributed Y_i, and $U(\beta_0) = \sum_i U_i(\beta_0)$, we have $\mathbb{E}[U(\beta_0)] = 0$ and $var[U(\beta_0)] = \sum_i \mathcal{I}_i(\beta_0) := \mathcal{I}(\beta_0)$. Under certain regulatory conditions, we can apply central limit theorem for independent yet non-identically distributed variables and yield:

$$\mathcal{I}^{-1/2}(\beta_0)U(\beta_0) \xrightarrow{d} N(0, \boldsymbol{I}_p) \tag{105}$$

Therefore, approximately

$$\mathcal{I}^{1/2}(\hat{\beta})(\hat{\beta} - \beta_0) \approx \mathcal{I}^{-1/2}(\hat{\beta})U(\beta_0) \xrightarrow{p} \mathcal{I}^{-1/2}(\beta_0)U(\beta_0) \xrightarrow{d} N(0, \boldsymbol{I}_p) \tag{106}$$

so we have the asymptotic distribution of $\hat{\beta}$:

$$\mathcal{I}^{1/2}(\hat{\beta})(\hat{\beta} - \beta_0) \xrightarrow{d} N(0, \boldsymbol{I}_p) \tag{107}$$

or more loosely,

$$\hat{\beta} \rightsquigarrow N(\beta_0, \mathcal{I}^{-1}(\hat{\beta})) \tag{108}$$

where we write $\rightsquigarrow$ for "asymptotically distributed as".

Based on Equation (108), we may now construct tests and confidence intervals for any linear combination for β. Most generally, suppose we are interested in $L\beta_0$, where L is an $m \times p$ matrix of rank m, then the asymptotic distribution for $L\hat{\beta}$ under the null hypothesis would be

$$L\hat{\beta} \rightsquigarrow N(L\beta_0, L\mathcal{I}^{-1}(\hat{\beta})L^\top) \tag{109}$$

Consequently, we have

$$(L\hat{\beta} - L\beta_0)^\top (L\mathcal{I}^{-1}(\hat{\beta})L^\top)^{-1}(L\hat{\beta} - L\beta_0) \xrightarrow{d} \chi^2_m \tag{110}$$

Based on Equation (110), we may construct asymptotic tests regarding to $L\beta_0$. For example, suppose we would like to test the null hypothesis $L\beta_0 = c$ where c is a constant vector, then we may calculate the following statistic

$$W := (L\hat{\beta} - c)^\top (L\mathcal{I}^{-1}(\hat{\beta})L^\top)^{-1}(L\hat{\beta} - c) \tag{111}$$

which has an asymptotic null distribution of χ^2_m. Since W will be larger as $L\hat{\beta}$ is farther from c, we can reject H_0 when $W > \chi^2_{m,1-\alpha}$, where α is the significance level. This is called the *Wald test*, based on the work of statistician Abraham Wald.

As a special case, if L is a row vector, then based on Equation (109),

$$\frac{L\hat{\beta} - L\beta_0}{\sqrt{L\mathcal{I}^{-1}(\hat{\beta})L^\top}} \xrightarrow{d} N(0, 1) \tag{112}$$

So to test the null hypothesis $H_0 : L\beta_0 = c$, we may calculate the follow Wald statistic:

$$W := \frac{L\hat{\beta} - c}{\sqrt{L\mathcal{I}^{-1}(\hat{\beta})L^\top}} \tag{113}$$

and reject H_0 when $|W|$ exceeds $Z_{1-\alpha/2}$, or output a p-value of $2 \times (1 - \Phi(|W|))$. In addition, based on Equation (112), we have

$$\mathbb{P}\left(-Z_{1-\alpha/2} < \frac{L\hat{\beta} - L\beta_0}{\sqrt{L\mathcal{I}^{-1}(\hat{\beta})L^\top}} < Z_{1-\alpha/2}\right) \xrightarrow{p} 1 - \alpha \tag{114}$$

So we may construct the following confidence interval for asymptotic $(1 - \alpha)$ coverage of $L\beta_0$

$$(L\hat{\beta} - Z_{1-\alpha/2}\sqrt{L\mathcal{I}^{-1}(\hat{\beta})L^\top}, L\hat{\beta} + Z_{1-\alpha/2}\sqrt{L\mathcal{I}^{-1}(\hat{\beta})L^\top}) \tag{115}$$

In most commercial packages, the confidence intervals and p-values for regression coefficients are calculated with Wald-type statistics / pivotal quantities. However, most packages (including R) will not explicitly list out $\mathcal{I}^{-1}(\hat{\beta})$, so to do inference on linear combinations of coefficients, some manual work will be needed most of the time.

6.2 Model comparison with likelihood ratio test

In the previous subsection, we approximated the score function $U(\beta_0)$ using Taylor expansion starting from the MLE $\hat{\beta}$. We can also do second-order Taylor expansion of the likelihood function $\ell(\beta_0)$ and yield

$$\ell(\beta_0) \approx \ell(\hat{\beta}) + \ell'^\top(\hat{\beta})(\beta_0 - \hat{\beta}) + \frac{1}{2}(\beta_0 - \hat{\beta})^\top \ell''(\hat{\beta})(\beta_0 - \hat{\beta}) \tag{116}$$

$$= \ell(\hat{\beta}) + \frac{1}{2}(\beta_0 - \hat{\beta})^\top \ell''(\hat{\beta})(\beta_0 - \hat{\beta}) \tag{117}$$

$$= \ell(\hat{\beta}) - \frac{1}{2}(\beta_0 - \hat{\beta})^\top I(\hat{\beta})(\beta_0 - \hat{\beta}) \tag{118}$$

$$\approx \ell(\hat{\beta}) - \frac{1}{2}(\hat{\beta} - \beta_0)^\top \mathcal{I}(\hat{\beta})(\hat{\beta} - \beta_0) \tag{119}$$

where Equation (117) is true since $\hat{\beta}$ is found by letting the derivative of the log-likelihood function be zero, Equation (118) is using the definition of observed information and Equation (119) approximates the observed information with Fisher information. Rearranging the above and we get

$$(2\ell(\hat{\beta}) - 2\ell(\beta_0)) \approx (\hat{\beta} - \beta_0)^\top \mathcal{I}(\hat{\beta})(\hat{\beta} - \beta_0) \xrightarrow{d} \chi^2_p \tag{120}$$

where the last converging distribution comes from setting $L = \boldsymbol{I}_p$ in Equation (110). This results also gives us an alternative asymptotic test for $H_0 : \beta = \beta_0$. When H_0 is not true, $\hat{\beta}$ would give a vastly better fit than β_0, reflected by a large value of $2\ell(\hat{\beta}) - 2\ell(\beta_0)$. So we may reject H_0 when $2\ell(\hat{\beta}) - 2\ell(\beta_0) > \chi^2_{p,1-\alpha}$, where α is the significance level. Note that

$$2\ell(\hat{\beta}) - 2\ell(\beta_0) = 2\log\frac{L(\hat{\beta})}{L(\beta_0)} \tag{121}$$

where L is the likelihood function, so this test statistic is actually related to the ratio of likelihoods. Therefore, we term it as the *(log-)likelihood ratio statistic*, and the test is called *likelihood ratio test* (LRT).

Another way to look at the (log-)likelihood ratio statistic is that we are fitting two models: one with β freely estimated by maximum likelihood (larger model), and one with β fixed to β_0 with no free parameters left (smaller model). The log-likelihood difference between these two models are subtracted and timed by 2, which under the null hypothesis that the smaller model is correct, should be asymptotically distributed as χ^2_p, where p is the difference of number of parameters between these two models. This concept of comparing the (log-)likelihood of nested models can be generalized by a phenomenal theorem put forward by Samuel Wilks, which we elaborate below.

Suppose that we have two nested candidate models M_1 and M_0. M_1 is the "larger" model with parameter vector θ of dimension p. M_0 is the "smaller" model that is a special case of M_1 by adding constraints on θ so that the remaining degrees of freedom for the parameters is q (of course $q < p$). Now we use maximum likelihood to estimate both models and calculate their log-likelihood by plugging in the parameter estimates. Then Wilks' theorem tells us that under certain regulatory conditions,

$$\Lambda := 2(\ell_{M_1}(\hat{\theta}_{M_1}) - \ell_{M_0}(\hat{\theta}_{M_0})) \xrightarrow{d} \chi^2_{p-q} \tag{122}$$

In other words, if the difference of number of parameters between two nest models is ν, then twice the log-likelihood difference between them would asymptotically follow χ^2_ν under the null hypothesis that the smaller model is correct. When the null hypothesis is *incorrect*, implying the larger model has way better bit than the smaller model, the log-likelihood ratio statistic should be large. Therefore, we reject the null hypothesis when the statistic exceeds $\chi^2_{\nu,1-\alpha}$, where α is the significance level.

LRTs are used extensively in model selection. In statistical modelling, we strive to find a succinct model that has adequate fit. If two models have similar fit to the data but one has less parameters than the other, then by the *Occam's razor* rule, we would opt for the one with less parameters. In LRT, when the null hypothesis is rejected, we would of course choose the larger model since it has better fit. However, when we fail to reject the null hypothesis, then we would consider choosing the small model, since it is more concise and we do not have evidence that it performs worse than the larger model.

Note that now we have both LRT and Wald test at hand, we have two different strategies that can test if a subvector θ^* of our parameter vector θ equals to a predetermined value (i.e. $H_0 : \theta^* = c$). One is to fit a model M_1 estimating the whole θ vector, and use Wald test (Equation (111)) to directly test if $\theta^* = c$. Another is to fit another model M_0 that constrains the subvector θ^* to be c and use LRT between M_1 and M_0. These two approach are asymptotically equivalent, but extensive simulation studies have shown that in finite samples, LRT performs better than Wald test, and it is also harder to carry out multivariate Wald test in commercial packages. Therefore, for this type of test, LRT is almost always preferred.

6.3 Deviance for goodness-of-fit

Data modelling is a continous process. At times we would like to know if our model adequately fits the data, so that we could modify our model if the fit is not satisfactory. Therefore, we require a statistic that measures the difference of degree of fit between our current model and the saturated model (we will talk about its definition later). When the current model has significantly worse fit than the saturated model, we deem that the current model does not fit well and model modification

is needed. In contrast, when the difference in fit is non-significant between the current model and the saturated model, we would at least some confidence that the model is doing its job.

In GLMs, one way to quantify goodness-of-fit (or badness-of-fit) is the *deviance* statistic, which is just the log-likelihood ratio statistic comparing the current model and the saturated model. Before we lay out the formula for deviance, we need to define what a saturated model is. Recall the basic structure of GLMs:

$$Y_i|X_i \sim f(\theta_i, \phi) \tag{123}$$

$$g(\mu_i) := \eta_i = X_i\beta \tag{124}$$

Here each observation has its own θ_i, but θ_i is not freely estimated – it is restricted by its relation to β. Therefore, most of the time $\boldsymbol{Y}_i$ would *not* be identical to the predicted $\mathbb{E}[Y_i|X_i] = g^{-1}(X_i\hat{\beta})$. In the saturated model, however, we insert as many free θ_i as possible (one for each value of X_i). For example, suppose $Y_i|X_i$ follows a Poisson distribution and X_i is supposed to differ between observations, then each observation would have a rate parameter λ_i. Since $\mathbb{E}[Y_i|X_i] = \lambda_i$, the estimated λ_i in the saturated model would just be $\hat{\lambda}_{(\text{sat})i} = \boldsymbol{Y}_i$. The log-likelihood contributed by observation i would then be

$$\ell_{(\text{sat})i} = \boldsymbol{Y}_i \log \hat{\lambda}_{(\text{sat})i} - \hat{\lambda}_{(\text{sat})i} - \log \boldsymbol{Y}_i! = \boldsymbol{Y}_i \log \boldsymbol{Y}_i - \boldsymbol{Y}_i - \log \boldsymbol{Y}_i! \tag{125}$$

By convention, when $\boldsymbol{Y}_i = 0$ we let $0 \log 0 = 0$. The total log-likelihood for the saturated model would then be

$$\ell_{(\text{sat})} = \sum_i (\boldsymbol{Y}_i \log \boldsymbol{Y}_i - \boldsymbol{Y}_i - \log \boldsymbol{Y}_i!) \tag{126}$$

Here we can see that since each observation has its own parameter λ_i, we have a total of n parameters, which is the maximal number of parameters we can put into a data of size n, hence the name *saturated* model. However, note that in the case where the predictors are categorical, the saturated model may not have n parameters since there are limited possible value combinations for X. For example, if X only contains three dummy variables coding a three-category grouping, then there are only three possible value configurations for X, so the number of parameters for the saturated model would be 3.

Now suppose using maximum likelihood, our current model estimates the rate parameter for observation i is $\hat{\lambda}_i$, then the total log-likelihood for the model is

$$\ell = \sum_i (\boldsymbol{Y}_i \log \hat{\lambda}_i - \hat{\lambda}_i - \log \boldsymbol{Y}_i!) \tag{127}$$

So we have the LRT statistic comparing our model and the saturated model as

$$D = 2(\ell_{(sat)} - \ell) = 2\sum_i \Big[\boldsymbol{Y}_i \log\Big(\frac{\boldsymbol{Y}_i}{\hat{\lambda}_i}\Big) - (\boldsymbol{Y}_i - \hat{\lambda}_i)\Big] = 2\Big[\sum_i \boldsymbol{Y}_i \log\Big(\frac{\boldsymbol{Y}_i}{\hat{\lambda}_i}\Big) - \sum_i (\boldsymbol{Y}_i - \hat{\lambda}_i)\Big] \tag{128}$$

In GLMs with intercepts, the grand mean of outcome is well-calibrated, so that $\frac{1}{n}\sum_i \boldsymbol{Y}_i = \frac{1}{n}\sum_i \hat{\mu}_i$. In this case, the LRT statistics can be further reduced to

$$D = 2\sum_i \boldsymbol{Y}_i \log\Big(\frac{\boldsymbol{Y}_i}{\hat{\lambda}_i}\Big) \tag{129}$$

Under the null hypothesis that our current model fits the data well, D should have an asymptotic distribution of χ^2_{m-p} based on the LRT theory, where m and p are the numbers of parameters in the saturated model and our current model. As the lack of fit for our model becomes greater, D would be larger (since the log-likelihood for our model would be smaller), so we reject the null hypothesis and claim our model does not fit the data well when $\Lambda > \chi^2_{m-p,1-\alpha}$, where α is the significance level.

Note that LRT between two models will sometimes be framed as comparison between the deviance of the models. Suppose a larger model 1 and smaller model 2 have log-likelihood ℓ_1, ℓ_2 and deviance D_1, D_2, then we have the relation

$$D_1 = 2(\ell_{\text{sat}} - \ell_1) \tag{130}$$

$$D_2 = 2(\ell_{\text{sat}} - \ell_2) \tag{131}$$

So the log-likelihood ratio statistic comparing the two models would be $\Lambda = 2(\ell_1 - \ell_2) = D_2 - D_1$. That is, the statistic can be seen as subtracting the deviance of the larger model from the smaller model. The null distribution of the statistic remains the same, which is a chi-squared distribution with degrees of freedom equal to the difference of number of parameters between the two models.

7 Generalized Linear Models of Binary Outcomes

We have gone through the general form of GLMs, along there estimation and inference procedures. Now we will look specifically into models with binary outcomes, which can be expressed as

$$Y_i|X_i \sim \text{Bernoulli}(p_i) \tag{132}$$
$$g(p_i) = X_i\beta \tag{133}$$

In the following, we will assume that the link function $g(.)$ is chosen to be the canonical link, logit, so that the model becomes,

$$Y_i|X_i \sim \text{Bernoulli}(p_i) \tag{134}$$
$$\log \frac{p_i}{1-p_i} = X_i\beta \tag{135}$$

Suppose we can express X as a row vector of covariates $(x_0 = 1, x_1, x_2, ..., x_{p-1})$ and β as the column vector of regression coefficient $(\beta_0, \beta_1, \beta_2, ..., \beta_{p-1})^\top$, the we have

$$\log \frac{p}{1-p} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_{p-1} x_{p-1} \tag{136}$$

This expression grants us an explanation for the regression coefficients. Note that the left-hand side, $\log \frac{p}{1-p}$, is the *log odds* of event, so the difference in values of $X\beta$ between two scenarios reflects the event *log odds ratio* between the scenarios:

- β_0: The right-hand side becomes β_0 when $x_1, x_2, ..., x_{p-1}$ are all plugged in as 0. Therefore, β_0 is the event log odds under the scenario where all covariates has value 0.
- β_k: If we increase x_k by 1 unit while holding other covariates as constant, the right-hand side would increase by β_k, i.e. the event log odds ratio would be β_k. In other words, the event odds would become $\exp(\beta_k)$ times the original odds everytime x_k is increased by 1.

More often than not, we will be interested in the inference of the β_k's: if $\beta_k \neq 0$, then changes in x_k would influence the event probability, meaning that x_k plays a role in explaining the outcome of interest. Tests for β_k can be carried out with Wald test or likelihood ratio test (LRT), with the former directly modelling the asymptotic distribution of $\hat{\beta}_k$, and the latter comparing two models where one estimates β_k freely and one constrains β_k to be zero. For confidence interval construction, it is practically more common present the interval for *odds ratios* rather than for log odds ratios. Therefore, after obtaining the confidence interval for the log odds ratio β_k as $(\hat{l}_{\beta_k}, \hat{u}_{\beta_k})$, we may convert it to the confidence interval for the odds ratio $\exp(\beta_k)$ as $(\exp(\hat{l}_{\beta_k}), \exp(\hat{u}_{\beta_k}))$. When the confidence interval of $\exp(\beta_k)$ does not cover 1, it implies that x_k influences the outcome. In the following, we will show how the methods above can be used in applied analyses.

7.1 Logistic Regression with Limited Covariate Patterns

We first consider a special scneario where we try to model our binary outcome over a single categorical variable with K levels. In this case, we may divide the observations into K subgroups based on their values for the categorical variable, shown in Table 2.

	Subgroups				Total
	1	2	$\cdots$	K	
Events	y_1	y_2	$\cdots$	y_K	y
Non-events	$n_1 - y_1$	$n_2 - y_2$	$\cdots$	$n_N - y_K$	$n - y$
Total	n_1	n_2	$\cdots$	n_K	n

Table 2: Contingency table representation of binary outcome data with categorical predictors

Note that here we have *aggregated* the data, so y_i now stands for the number of events for subgroup i and *not* the observed outcome for individual i. A most commonly asked question for this type of data is whether all subgroups have the same event rate, i.e. testing for the following set of hypotheses:

H_0 : All subgroups have the same event probability

H_1 : At least one subgroup has different event probability than others

This set of hypotheses is the same as testing whether a model setting identical event probability for each subgroup fits the data well, which can be answered by goodness-of-fit test using deviance. To obtain the deviance, we first need to know what the "saturated model" is. We said that a saturated model is a model which assigns a distinct parameter θ_i for each *covariate pattern*. For this piece of data, our saturated model would have K event probability parameters $(p_1, p_2, ..., p_K)^\top$, one for each subgroup. The saturated model can then be written as

$$y_i \sim \text{Binomial}(p_i, n_i) \tag{137}$$

with log-likelihood function

$$\sum_{i=1}^{K} \Big[\log \binom{n_i}{y_i} + y_i \log p_i + (n_i - y_i) \log(1 - p_i) \Big] \tag{138}$$

which upon maximization with respect to $(p_1, p_2, ..., p_K)^\top$ would yield the estimates

$$\hat{p}_i = \frac{y_i}{n_i} \tag{139}$$

Therefore, the observed log-likelihood for the saturated model can be written as

$$\ell_{\text{sat}} = \sum_{i=1}^{K} \Big[\log \binom{n_i}{y_i} + Y_i \log \hat{p}_i + (n_i - y_i) \log(1 - \hat{p}_i) \Big] \tag{140}$$

$$= \sum_{i=1}^{K} \Big[\log \binom{n_i}{y_i} + y_i \log \frac{y_i}{n_i} + (n_i - y_i) \log \frac{n_i - y_i}{n_i} \Big] \tag{141}$$

$$= \sum_{i=1}^{K} \Big[y_i \log y_i + (n_i - y_i) \log(n_i - y_i) \Big] + \sum_{i=1}^{K} \Big[\log \binom{n_i}{y_i} - n_i \log n_i \Big] \tag{142}$$

Under H_0, all subgroups have the same event probability p, so the model becomes

$$y_i \sim \text{Binomial}(p, n_i) \tag{143}$$

with log-likelihood function

$$\sum_{i=1}^{K} \Big[\log \binom{n_i}{y_i} + y_i \log p + (n_i - y_i) \log(1 - p) \Big] \tag{144}$$

$$= \big(\sum_{i=1}^{K} y_i \big) \log p + \big(\sum_{i=1}^{K} (n_i - y_i) \big) \log(1 - p) + \sum_{i=1}^{K} \log \binom{n_i}{y_i} \tag{145}$$

$$= y \log p + (1 - y) \log(1 - p) + \sum_{i=1}^{K} \log \binom{n_i}{y_i} \tag{146}$$

which upon maximization yields the straightforward estimate for p

$$\hat{p} = \frac{y}{n} \tag{147}$$

And the predicted number of events for subgroup i can be written as

$$\hat{y}_i = n_i \hat{p} \tag{148}$$

Therefore, the observed log-likelihood for the model under H_0 can be written as

$$\ell = \sum_{i=1}^{K} \Big[\log \binom{n_i}{y_i} + y_i \log \hat{p} + (n_i - y_i) \log(1 - \hat{p}) \Big] \tag{149}$$

$$= \sum_{i=1}^{K} \Big[\log \binom{n_i}{y_i} + y_i \log \frac{n_i \hat{p}}{n_i} + (n_i - y_i) \log \frac{n_i - n_i \hat{p}}{n_i} \Big] \tag{150}$$

$$= \sum_{i=1}^{K} \Big[\log \binom{n_i}{y_i} + y_i \log \frac{\hat{y}_i}{n_i} + (n_i - y_i) \log \frac{n_i - \hat{y}_i}{n_i} \Big] \tag{151}$$

$$= \sum_{i=1}^{K} \Big[Y_i \log \hat{y}_i + (n_i - y_i) \log(n_i - \hat{y}_i) \Big] + \sum_{i=1}^{K} \Big[\log \binom{n_i}{y_i} - n_i \log n_i \Big] \tag{152}$$

From Equations (142) and (152), we have the deviance statistic:

$$D = 2(\ell_{\text{sat}} - \ell) \tag{153}$$

$$= 2\sum_{i=1}^{K}\Big[y_i \log y_i + (n_i - y_i)\log(n_i - y_i)\Big] - \sum_{i=1}^{K}\Big[Y_i \log \hat{y}_i + (n_i - y_i)\log(n_i - \hat{y}_i)\Big] \tag{154}$$

$$= 2\sum_{i=1}^{K}\Big[y_i \log\big(\frac{y_i}{\hat{y}_i}\big) + (n_i - y_i)\log\big(\frac{n_i - y_i}{n_i - \hat{y}_i}\big)\Big] \tag{155}$$

which should be asymptotically distributed as χ^2_{K-1}, since the saturated model has K parameters and the model under H_0 has only 1 parameter.

Note that when H_0 is true, y_i should be close to $\hat{y}_i$, and $n - \hat{y}_i$ should be close to $n_i - \hat{y}_i$. Therefore, considering the following Taylor expansion for $f(x) = x\log\big(\frac{x}{x_0}\big)$ around $x = x_0$:

$$f(x) = x\log\big(\frac{x}{x_0}\big) \tag{156}$$

$$\approx f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2}(x - x_0)^2 \tag{157}$$

$$= 0 + 1\cdot(x - x_0) + \frac{1/x_0}{2}(x - x_0)^2 = (x - x_0) + \frac{1}{2}\frac{(x - x_0)^2}{x_0} \tag{158}$$

Using the above to approximate Equation (155) and we yield

$$D = 2\sum_{i=1}^{K}\Big[y_i \log\big(\frac{y_i}{\hat{y}_i}\big) + (n_i - y_i)\log\big(\frac{n_i - y_i}{n_i - \hat{y}_i}\big)\Big] \tag{159}$$

$$\approx 2\sum_{i=1}^{K}\left[(y_i - \hat{y}_i) + \frac{1}{2}\frac{(y_i - \hat{y}_i)^2}{\hat{y}_i} + [(n_i - y_i) - (n_i - \hat{y}_i)] + \frac{1}{2}\frac{[(n_i - y_i) - (n_i - \hat{y}_i)]^2}{n_i - \hat{y}_i}\right] \tag{160}$$

$$= \sum_{i=1}^{K}\left[\frac{(y_i - \hat{y}_i)^2}{\hat{y}_i} + \frac{[(n_i - y_i) - (n_i - \hat{y}_i)]^2}{n_i - \hat{y}_i}\right] \tag{161}$$

$$:= \sum_{j}\frac{(o_j - e_j)^2}{e_j} := X^2 \tag{162}$$

Here j rotates through each *cell* in the contingency table, where o_j and e_j are the observed and expected count for each cell under H_0. This is the well-known **Pearson chi-squared statistic** and should also have an asymptotic distribution of χ^2_{K-1}. In fact, simulations have shown that the Pearson chi-squared statistic X^2 actually performs *better* than the deviance D under finite samples since D can be largely influenced by small event frequencies. Nevertheless, goodness-of-fit test based on deviance usually suffices for applied analyses.

Now let us demonstrate how the model above can be carried out in the *glm* package in *R*. Suppose we have a set of data in contingency table form shown in Table 3. We may construct the data in *R* using the following code:

	Subgroups		
	A	*B*	*C*
Events	30	40	20
Non-events	60	40	60
Total	90	80	80

Table 3: Example contingency table of binary outcome data with categorical predictors

```
data_agg_1 <- data.frame(y1 = c(30, 40, 20),
                         y0 = c(60, 40, 60),
                         group = c("A", "B", "C"))
print(data_agg_1)
```

```
##   y1 y0 group
## 1 30 60     A
## 2 40 40     B
## 3 20 60     C
```

Now we try to fit a saturated model, i.e. a model that assumes each subgroup has possibly distinct even probability. This can be carried out by including the group variable into the model, which would end up in two dummy variables entering the model since the group variable has three levels (why?).

```
mod_agg_sat_1 <- glm(cbind(y1, y0) ~ factor(group),
                     family = binomial(), data = data_agg_1)
summary(mod_agg_sat_1)

##
## Call:
## glm(formula = cbind(y1, y0) ~ factor(group), family = binomial(),
##     data = data_agg_1)
##
## Deviance Residuals:
## [1]  0  0  0
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -0.6931     0.2236  -3.100  0.00194 **
## factor(group)B   0.6931     0.3162   2.192  0.02839 *
## factor(group)C  -0.4055     0.3416  -1.187  0.23520
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1.1259e+01  on 2  degrees of freedom
## Residual deviance: 4.4409e-15  on 0  degrees of freedom
## AIC: 20.235
##
## Number of Fisher Scoring iterations: 3
```

We provide some explanations of the code and output:

- In the code calling the function `glm`, `cbind(y1, y0)` assigns a two-column matrix as the outcome variable, which indicates that the data is in aggregate form and `y1` stands for number of events and `y0` stands for number of non-events. `factor(group)` adds the group categorical variable into the predictors. `family = binomial()` specifics the outcome distribution as binary. Note that we did not specify the link function here, so the function chooses the default link for binary outcomes, logit. You'll have to specify `binomial("probit")` or `binomial("cloglog")` if you want to use probit link or complementary log-log link instead.
- In the summary of results, we see that there are three lines standing for three regression coefficients: `(Intercept)`, `factor(group)B` and `factor(group)C`. That is, the log odds is modelled as:
$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \mathbf{1}(\text{group B})\beta_1 + \mathbf{1}(\text{group C})\beta_2 \tag{163}$$
Therefore, $\hat{\beta}_0 =$ `-0.6931` is the estimated event log odds for group A. $\hat{\beta}_1 =$ `0.6931` is the estimated event log odds ratio of group B compared to group A. $\hat{\beta}_2 =$ `-0.4055` is the estimated event log odds ratio of group C compared to group A.
- The `Pr(>|z|)` column is the Wald test for the null hypothesis that each regression coefficient is zero. However, we would like to test if the regression coefficients for the two factor dummy variables are *both* zero, and this output does not give us information for this test.
- In the deviance part, we first see the null deviance, which is the deviance of a model with *only the intercept*, and therefore has a degree of freedom of 2. The residual deviance is the deviance of the current model, which should be zero (with some rounding error) since this model is exactly the saturated model.

In the output above, the regression coefficient estimates (for log odds ratios) are given, yet most of the time we would want the odds ratio estimates and their confidence intervals. This can be produced with the following code:

```r
exp(coef(mod_agg_sat_1))
```

```
##    (Intercept) factor(group)B factor(group)C
##      0.5000000      2.0000000      0.6666667
```

```r
exp(confint.default(mod_agg_sat_1, level = 0.95))
```

```
##                     2.5 %    97.5 %
## (Intercept)     0.3225786 0.7750049
## factor(group)B 1.0761094 3.7170941
## factor(group)C 0.3413250 1.3021152
```

Here, the `confint.default` function outputs Wald confidence intervals. Using the `confint` instead of the `confint.default` function would produce likelihood-ratio based confidence intervals, which should be similar to Wald intervals asymptotically. Based on the output, we have the event odds ratio between group B and group A estimated to be `2.00` with 95% confidence interval `(1.076 ~ 3.717)`, which does not include 1 and corresponds to the significance of Wald test for the regression coefficient shown previously. The event odds ratio between group C and group A is estimated to be `0.67` with 95% confidence interval `(0.341 ~ 1.302)`.

Now we set out to test if the three subgroups have identical event probabilities. We first use LRT, which involves comparing the likelihood of the model above with a model that gives all groups the same event probabilities, which we fit bellow:

```r
mod_agg_null_1 <- glm(cbind(y1, y0) ~ 1, family = binomial(), data = data_agg_1)
summary(mod_agg_null_1)
```

```
##
## Call:
## glm(formula = cbind(y1, y0) ~ 1, family = binomial(), data = data_agg_1)
##
## Deviance Residuals:
##       1        2        3
## -0.5301   2.5557  -2.1088
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.5754     0.1318  -4.367 1.26e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 11.259  on 2  degrees of freedom
## Residual deviance: 11.259  on 2  degrees of freedom
## AIC: 27.494
##
## Number of Fisher Scoring iterations: 4
```

The `~ 1` in the `glm` function indicates that we want a model with only the intercept, so in the output we only have one regression coefficient. The log odds is now modelled as

$$\log\left(\frac{p}{1-p}\right) = \beta_0 \tag{164}$$

Also note the null deviance is identical to the residual deviance, which reconfirms that we are fitting a model with only the intercept. With both models at hand, we can conduct the LRT with the `anova` function:

```r
anova(mod_agg_null_1, mod_agg_sat_1, test = "LRT")
```

```
## Analysis of Deviance Table
##
```

```
## Model 1: cbind(y1, y0) ~ 1
## Model 2: cbind(y1, y0) ~ factor(group)
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1         2     11.259
## 2         0      0.000  2   11.259  0.00359 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can see that the LRT has degrees of freedom of 2, which is the difference of 3 parameters in the saturated model and 1 parameter in the model under the null hypothesis. The test was significant under significance level 0.05 with p-value `0.00359`, so we may conclude that the event probabilities of the three subgroups are *not* identical. Alternatively, we can use the goodness-of-fit test based on Pearson chi-squared statistics:

```
chisq.test(data_agg_1[, c("y1", "y0")])

##
##  Pearson's Chi-squared test
##
## data:  data_agg_1[, c("y1", "y0")]
## X-squared = 11.285, df = 2, p-value = 0.003544
```

Here in the code we input the first two columns of our data, i.e. the contingency table into `chisq.test` to obtain the results for Pearson's chi-squared test for goodness-of-fit. The X^2 statistic is `11.285`, which is actually very close to the LRT statistics, `11.259`. The degrees of freedom is of course also 2, and we have a very similar p-value `0.00354`. Still another way to carry out the test is to simultaneously test if β_1 and β_2 in Equation (163) are simultaneously zero with Wald test. However, this kind of test is not included in default packages in R, so we will have to calculate the test statistic manually:

```
(fisher_inverse <- summary(mod_agg_sat_1)$cov.scaled)

##                 (Intercept) factor(group)B factor(group)C
## (Intercept)            0.05          -0.05     -0.0500000
## factor(group)B        -0.05           0.10      0.0500000
## factor(group)C        -0.05           0.05      0.1166667

(L <- matrix(c(0,1,0,0,0,1), 2, 3, byrow = T))

##      [,1] [,2] [,3]
## [1,]    0    1    0
## [2,]    0    0    1

(beta_hat <- coef(mod_agg_sat_1))

##    (Intercept) factor(group)B factor(group)C
##     -0.6931472      0.6931472     -0.4054651

W <- t(L %*% beta_hat) %*%
       solve(L %*% fisher_inverse %*% t(L)) %*%
       (L %*% beta_hat)
1-pchisq(W, 2)

##             [,1]
## [1,] 0.004139626
```

Here `summary(mod_agg_sat_1)$cov.scaled` extracts the estimated variance-covariance matrix of the regression coefficient estimator (which includes intercept so is of dimension 3×3), i.e. the estimated inverse of Fisher information. `L` is the linear transformation matrix we mentioned in the previous section. `beta_hat` is the vector of regression coefficient estimates. We can then construct the Wald statistic using matrix and compare it to the upper tail of χ^2_2. The degrees of freedom is 2 since L is a rank-2 matrix. The resulting p-value is `0.00414`, which is little bit different from LRT and Pearson chi-squared test, but not too far off.

Question: How do we write the R code to test if group B and C have the same event probability?

Alternatively, we may use individual observation data to carry out the analysis:

```
# Prepare individual observation data
data_ind_1 <- data_agg_1 %>%
  pivot_longer(y1:y0, names_to = "y", names_prefix = "y",
               names_transform = list(y = as.integer),
               values_to = "count") %>%
  uncount(count)

head(data_ind_1)

## # A tibble: 6 x 2
##   group     y
##   <chr> <int>
## 1 A         1
## 2 A         1
## 3 A         1
## 4 A         1
## 5 A         1
## 6 A         1

mod_ind_sat_1 <- glm(y ~ factor(group), family = binomial(), data = data_ind_1)
mod_ind_null_1 <- glm(y ~ 1, family = binomial(), data = data_ind_1)

summary(mod_ind_sat_1)

##
## Call:
## glm(formula = y ~ factor(group), family = binomial(), data = data_ind_1)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1774  -0.9005  -0.7585   1.1774   1.6651
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -0.6931     0.2236  -3.100  0.00194 **
## factor(group)B   0.6931     0.3162   2.192  0.02838 *
## factor(group)C  -0.4055     0.3416  -1.187  0.23520
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 326.71  on 249  degrees of freedom
## Residual deviance: 315.45  on 247  degrees of freedom
## AIC: 321.45
##
## Number of Fisher Scoring iterations: 4

exp(coef(mod_ind_sat_1))

##    (Intercept) factor(group)B factor(group)C
##      0.5000000      2.0000000      0.6666667

exp(confint.default(mod_ind_sat_1, level = 0.95))

##                    2.5 %    97.5 %
## (Intercept)    0.3225786 0.7750049
## factor(group)B 1.0761095 3.7170939
## factor(group)C 0.3413251 1.3021147
```

```r
anova(mod_ind_null_1, mod_ind_sat_1, test = "LRT")

## Analysis of Deviance Table
##
## Model 1: y ~ 1
## Model 2: y ~ factor(group)
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       249     326.71
## 2       247     315.45  2   11.259  0.00359 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

7.2 Logistic Regression with Continuous Covariates

In the case where X contains continuous variables, it is often the case that the number of covariate patterns is very close to the number of observations (unless you are willing to bin the continuous variables). Therefore, we would have to rely on models on means, i.e. $g(\mu_i) = X_i\beta$ to reduce the number of parameters. The estimation and inference procedure remains similar to that of limited covariate patterns, but with one caveat: the goodness-of-fit test based on deviance or Pearson chi-squared statistic does not perform well when the number of covariate patterns is large, because the expected frequencies of the contingency table would be small, and the approximations will tend to be poor.

The most classic approach for this scenario is the **Hosmer-Lemeshow test**, which groups observations into categories based on their predicted probabilities, and then use Pearson chi-squared test to check for goodness-of-fit. Typically, about $g = 10$ groups are used with approximately equal numbers of observations in each group. The data aggregated by the groups would then look like Table 2 where $K = g$. A Pearson chi-squared statistic X^2 can then be calculated, where simulations have shown that asymptotically, under the null hypothesis of the model fitting the data well, $X^2 \sim \chi^2_{g-2}$. Rejecting the null hypothesis indicates evidence that the model is ill-fitting.

In the following we will demonstrate how the approach works with an example dataset, where we aim to predict whether or not a patient has diabetes using their age (`age`), 2-hour blood glucose level after a oral glucose tolerance test (`glucose`), body mass index (`mass`) and the number of times they have been pregnant (`pregnant`):

```r
data("PimaIndiansDiabetes2", package = "mlbench")
head(PimaIndiansDiabetes2)

##   pregnant glucose pressure triceps insulin mass pedigree age diabetes
## 1        6     148       72      35      NA 33.6    0.627  50      pos
## 2        1      85       66      29      NA 26.6    0.351  31      neg
## 3        8     183       64      NA      NA 23.3    0.672  32      pos
## 4        1      89       66      23      94 28.1    0.167  21      neg
## 5        0     137       40      35     168 43.1    2.288  33      pos
## 6        5     116       74      NA      NA 25.6    0.201  30      neg

# Extract complete cases and relabel diabetes as 0 and 1
complete <- complete.cases(PimaIndiansDiabetes2[,
                c("diabetes", "age", "glucose", "mass", "pregnant")])
data_cont <- PimaIndiansDiabetes2[complete,]
data_cont$diabetes <- 0 + (data_cont$diabetes == "pos")
mod_cont <- glm(diabetes ~ age + glucose + mass + pregnant,
                family = binomial(), data = data_cont)
summary(mod_cont)

##
## Call:
## glm(formula = diabetes ~ age + glucose + mass + pregnant, family = binomial(),
##     data = data_cont)
##
## Deviance Residuals:
```

```
##     Min       1Q   Median       3Q      Max
## -2.2724  -0.7191  -0.4076   0.7363   2.3699
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -9.036125   0.718060 -12.584  < 2e-16 ***
## age          0.012015   0.009245   1.300 0.193713
## glucose      0.036183   0.003514  10.298  < 2e-16 ***
## mass         0.091207   0.014632   6.233 4.56e-10 ***
## pregnant     0.109336   0.031931   3.424 0.000617 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 974.75  on 751  degrees of freedom
## Residual deviance: 712.90  on 747  degrees of freedom
## AIC: 722.9
##
## Number of Fisher Scoring iterations: 5
```

Notice here that if we focus on the odds ratio estimate for the variable `glucose`:

```
exp(coef(mod_cont))["glucose"]

##  glucose
## 1.036846

exp(confint.default(mod_cont)["glucose",])

##    2.5 %   97.5 %
## 1.029730 1.044011
```

This implies that each unit (mg/dL) increase of blood glucose level would be projected to increase the odds of by 3.68% (95% confidence interval: (2.93% $\sim$ 4.31%)). However, this may not be useful since a 1 mg/dL change in blood glucose is immaterial clinically. A more impactful way would be to report how the odds would increase per 10 mg/dL increase in blood glucose level. This can be obtained by:

```
exp(coef(mod_cont))["glucose"]^10

##  glucose
## 1.435959

exp(confint.default(mod_cont)["glucose",])^10

##    2.5 %   97.5 %
## 1.340394 1.538336
```

That is, the odds would be increased by 43.60% (95% confidence interval: (34.04% $\sim$ 53.83%)) per 10 mg/dL increase in blood glucose level. Note that although the numbers have gone larger, the p-value for significance for the `glucose` would not change. Now let us try and use the Hosmer-Lemeshow test to determine if this model is ill-fitting:

```
# Calculate fitted probabilities and group the observations by quantiles
data_cont$fitted <- fitted.values(mod_cont, data_cont)
data_cont$group <- cut(data_cont$fitted,
                       quantile(data_cont$fitted, (0:10)/10),
                       include.lowest = T)
data_hl <- data_cont %>%
  group_by(group) %>%
  summarize(sum_prob = sum(fitted), sum_non_prob = sum(1-fitted),
```

```
            sum_event = sum(diabetes == 1), sum_non_event = sum(diabetes == 0))
data_hl

## # A tibble: 10 x 5
##    group          sum_prob sum_non_prob sum_event sum_non_event
##    <fct>             <dbl>        <dbl>     <int>         <int>
##  1 [0.0131,0.06]      3.17         72.8          0            76
##  2 (0.06,0.0964]      5.91         69.1          4            71
##  3 (0.0964,0.145]     9.29         65.7          7            68
##  4 (0.145,0.201]     12.9          62.1         15            60
##  5 (0.201,0.27]      17.5          57.5         23            52
##  6 (0.27,0.352]      23.2          51.8         29            46
##  7 (0.352,0.468]     29.8          45.2         28            47
##  8 (0.468,0.644]     41.7          33.3         39            36
##  9 (0.644,0.796]     54.1          20.9         54            21
## 10 (0.796,0.971]     66.4           9.61        65            11

hl <- sum((data_hl$sum_event - data_hl$sum_prob)^2/data_hl$sum_prob) +
  sum((data_hl$sum_non_event - data_hl$sum_non_prob)^2/data_hl$sum_non_prob)
hl

## [1] 10.23319

1-pchisq(hl, 8)

## [1] 0.249039
```

In the code above, `fitted.values` produces the predicted event probabilities of the original data based on the fitted model, whose quantiles are used to divide the observations into 10 groups. The fourth and fifth columns of `data_hl` is the contingency table constructed based on the 10 groups, and the second and third columns are the expected number of each cell in the contingency table calculated by summing the fitted probabilities of events (non-events) among observations within that cell. Finally, `hl` is the Peason chi-square type statistic, which is compared against a Chi-squared distribution with degrees of freedom of $10 - 2 = 8$. The test showed no evidence of the model being ill-fitting. A caution here is that Hosmer-Lemeshow test is known to be low-powered under small sample size, and its conclusion may depend largely on the number of groups g chosen.

7.3 Model Selection with Information Criteria

In the previous sections, we have demonstrated how to compare the fit between nested models with LRTs. However, these tests are designed to test if one model has significantly better fit than the other, but *not* to determine if one would have better predictive performance than the other. In addition, for model that are not nested, eg.

$$M_0 : g(\mu) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 M_1 : g(\mu) = \beta_0 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 \tag{165}$$

Analytically we have no asymptotic tests that compares the fit between M_0 and M_1, and we may nned to rely on parameteric bootstraps to really come up with a test.

If what we care about is predictive performance, then there are two commonly used criteria that can help use choose between models, even if they are not nested:

$$\text{Akaike Information Criterion (AIC): } -2\ell + 2p \tag{166}$$

$$\text{Bayesian / Schwartz Information Criterion (BIC / SIC): } -2\ell + p \log n \tag{167}$$

Here ℓ is the log-likelihood, p is the number of parameters and n is the number of observations. Intuitively, since we would want the fit to be better, we would want a model with larger ℓ, i.e. smaller -2ℓ. However, for a model with larger number of parameters, the chance of overfitting also increases. Therefore, AIC and BIC penalizes the number of parameters by a factor of 2 and $\log n$ so that the model selected would be less likely to exhibit overfit. In fact, *Mallow's* C_p in linear regression is a special case of AIC.

Comparison between usage of AIC and BIC has been done for a handful of models. A general consensus is that when the true model is within the pool of candidate models, then using BIC has

better chance of finding it. However, even if the true model is not within the pool of candidate models, AIC has the best chance of finding the *best approximating model*. There are also literature showing that AIC comparison is asymptotically equivalent to leave-one-out cross validation in linear regression.

8 Generalized Linear Models of Count Outcomes

We have seen how GLMs with binary outcomes, specifically logistic regressions, can be carried out with the `glm` package in *R*. We also showed that the Pearson Chi-square test for goodness of fit is actually asymptotically equivalent to likelihood ratio test on deviance. Now we turn our focus into models with count outcomes, which is modelled with Poisson distribution and its variants.

We first consider the case where the observation period for all observations are identical (say, 1 unit time). In this case, we can focus on modelling each observation's event rate λ_i, which is the *expected number of events within the unit time*. The model can be formally written as

$$Y_i|X_i \sim \text{Pois}(\lambda_i) \tag{168}$$
$$g(\lambda_i) = X_i\beta \tag{169}$$

In Poisson GLM, the default and canonical link function is $log(.)$, so the model becomes,

$$Y_i|X_i \sim \text{Pois}(\lambda_i) \tag{170}$$
$$\log \lambda_i = X_i\beta \tag{171}$$

As in logistic regression, we can express X as a row vector of covariates $(x_0 = 1, x_1, x_2, ..., x_{p-1})$ and β as the column vector of regression coefficient $(\beta_0, \beta_1, \beta_2, ..., \beta_{p-1})^\top$, then we have (omitting the i for brevity):

$$\log \lambda = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_{p-1} x_{p-1} \tag{172}$$

This expression grants us an explanation for the regression coefficients. On the left-hand side, $\log \lambda$, is the logarithm of the event rate, or in brief, *log rate*. Therefore, the difference in values of $X\beta$ between two scenarios reflects the event *log rate ratio* between the scenarios:

- β_0: The right-hand side becomes β_0 when $x_1, x_2, ..., x_{p-1}$ are all plugged in as 0. Therefore, β_0 is the log event rate under the scenario where all covariates has value 0.
- β_k: If we increase x_k by 1 unit while holding other covariates as constant, the right-hand side would increase by β_k, i.e. the event log rate ratio would be β_k. In other words, the event rate would become $\exp(\beta_k)$ times the original rate every time x_k is increased by 1.

The inference in Poisson regression is similar to that of logistic regression. If $\beta_k \neq 0$, or equivalently $\exp \beta_k \neq 1$, then changes in x_k would influence the event rate, so x_k plays a role in explaining the outcome of interest. Tests and confidence intervals for β_k and $\exp \beta_k$ can be carried out with Wald test or likelihood ratio test (LRT). Like logistic regression, for confidence intervals it is practically more common present the interval for *rate ratios* rather than for log rate ratios.

In practice, however, most of the time each subject is observed for different lengths of time (which in epidemiology is termed as the follow-up time). For two subjects of the same covariate values but different follow-up times, we would expect that their event rates (per unit time) would be similar, but the expected value of event number would not be the same. More precisely, since in Poisson distribution the event rate is assumed to be constant across time, the expected number of events should be proportional to the follow-up time. In the previous model, we assumed that for a subject with covariate value X_i, their event rate (per unit time) would be $\exp(X_i\beta)$. Now suppose the follow-up time for observation i is n_i units, then the expected number of events for observational i should be $n_i \exp(X_i\beta)$. Therefore, with heterogeneous follow-up time across subject, we would modify the model as follows

$$Y_i|X_i \sim \text{Pois}(\lambda_i) \tag{173}$$
$$\log \lambda_i = \log n_i + X_i\beta \tag{174}$$

Note that in Equation (174), the term $\log n_i$ is not the intercept: an intercept is a regression coefficient for the variable **1**, but here $\log n_i$ does not have any regression coefficients attached to it (or you may say that its regression coefficient is fixed to 1). This term is called the **offset**, implying it is an additional term that shifts the mean by a known amount. After adding the offset,

Table 9.1 *Deaths from coronary heart disease after 10 years among British male doctors categorized by age and smoking status in 1951.*

Age	Smokers		Non-smokers	
group	Deaths	Person-years	Deaths	Person-years
35–44	32	52407	2	18790
45–54	104	43248	12	10673
55–64	206	28612	28	5710
65–74	186	12663	28	2585
75–84	102	5317	31	1462

the explanation and inference of β remains similar. Now β_0 is the logarithm of the expected event number when *both* all covariate *and* $\log n_i$ is zero, meaning that β_0 is the log event rate in unit time.

In the following, we will show how the methods can be applied in *R*.

8.1 Poisson Regression by Example

As in data for binary outcome GLMs, in the case where the covariate pattern is limited, subjects with the same covariate values can be *aggregated*. In count outcome GLMs, the aggregated data would look like Table 4.

	Covariate pattern				Total
	1	2	$\cdots$	K	
Events	y_1	y_2	$\cdots$	y_K	y
Person-time	n_1	n_2	$\cdots$	n_K	n

Table 4: Contingency table representation of count outcome data with limited covariate patterns

Here *Events* and *Person-time* are the total number of events and total follow-up time within subjects of the corresponding covariate pattern. The concept of person-time is frequently used in life statistics, where the estimated event rate for a certain subpopulation can be estimated as

$$\text{Estimated event rate} = \frac{\text{Number of Events}}{\text{Follow-up Person-time}} \tag{175}$$

For example, if we are to estimate the acute attack rate in children with asthma aged 8 years old in year 2022, then we may gather the total number of recorded asthma attacks in 8-year-old children with asthma from the NHIRD, and divide it by the total person-time. Note that the follow-up time of each child would vary. For example, a child having 9-year-old birthday on Mar 1, 2023 would have only 3 months of follow-up, since they would be no long 8 years old after Mar 1.

Note that although Poisson regression are suitable for count data, i.e. events that can repeatedly occur, it can also handle event that can only occur once, eg. death. In this case, the subject is followed up until event occurs or they leave the study, and total number of events would be exactly the total number of subjects that experienced the event. This kind of data can also be analyzed with logistic regression by treating event as a binary outcome, however, in this case we are throwing away information from the follow-up person-time. For example, suppose the maximum follow-up time is one year, then a group of population dying within 3 months and dying between 3 to 6 months would be different in person-time in Poisson regression, but seen as the same in logistic regression (because they all died in these two scenario). Therefore, most of the time it is advised that for data with time-to-event information, Poisson regression (or better yet, survival analysis) should be preferred against logistic regression.

We will now investigate an example provided in the Dobson textbook Table 9.1 shown below. Suppose we would like to know:

- Is the death rate higher for smokers than non-smokers after adjusting for age (assuming that smoking has identical effect among each age group)?
- Is there differential effect of smoking related to age?

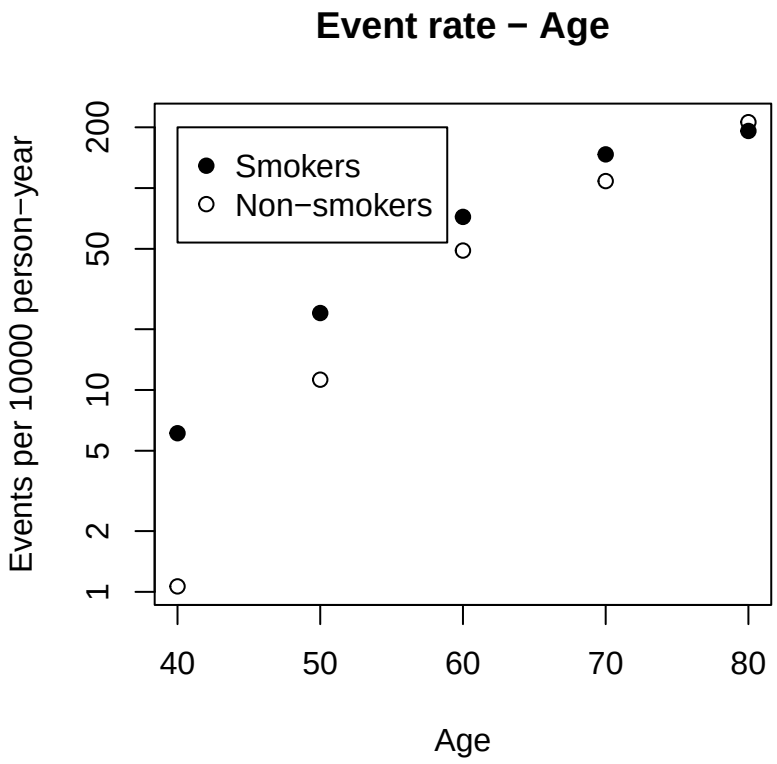
Note that this is aggregated data with subjects of the same smoking-age group combination pooled together. To simplify the analysis, we relabel the age group with the middle value, show in the code below,

```r
data <- data.frame(r = c(32, 104, 206, 186, 102, 2, 12, 28, 28, 31),
                   n = c(52407, 43248, 28612, 12663, 5317, 18790, 10673, 5710, 2585, 1462),
                   age = c(40, 50, 60, 70, 80, 40, 50, 60, 70, 80),
                   smoke = c(1, 1, 1, 1, 1, 0, 0, 0, 0, 0))
data
```

```
##       r     n age smoke
## 1    32 52407  40     1
## 2   104 43248  50     1
## 3   206 28612  60     1
## 4   186 12663  70     1
## 5   102  5317  80     1
## 6     2 18790  40     0
## 7    12 10673  50     0
## 8    28  5710  60     0
## 9    28  2585  70     0
## 10   31  1462  80     0
```

To assess if we should adjust for age linearly or non-linearly, we can plot

```r
plot(data$age, data$r / data$n * 10000, pch = ifelse(data$smoke == 1, 19, 1), log = "y",
     xlab = "Age", ylab = "Events per 10000 person-year", main = "Event rate - Age")
legend(40, 200, c("Smokers", "Non-smokers"), pch = c(19, 1))
```



From the graph, it looks like we can adjust for age using quadratic terms. Therefore, our model without interaction between age and smoking would be

```r
pois_mod <- glm(r ~ offset(log(n)) + smoke + age + I(age^2),
                family = poisson(), data = data)
summary(pois_mod)
```

```
##
## Call:
## glm(formula = r ~ offset(log(n)) + smoke + age + I(age^2), family = poisson(),
```

```
##     data = data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.14815  -0.71275   0.02995   0.33549   1.92253
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.787e+01  1.059e+00 -16.877  < 2e-16 ***
## smoke        3.545e-01  1.074e-01   3.302 0.000961 ***
## age          3.261e-01  3.426e-02   9.518  < 2e-16 ***
## I(age^2)    -1.944e-03  2.715e-04  -7.159 8.14e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 935.067  on 9  degrees of freedom
## Residual deviance:  12.176  on 6  degrees of freedom
## AIC: 75.243
##
## Number of Fisher Scoring iterations: 4
```

Note that we added a `offset(.)` over `log(n)` so that the function wouldn't misunderstand `log(n)` as a regular variable. For the family setting we use `poisson()`, so the link function would be the default and canonical link, natural log. As we can see, after adjusting for age non-linearly, the log rate ratio of smokers against non-smokers is positive, `0.3545`, which means that smoking seems to *increase* the rate of death. From the Wald test on the right, this log rate ratio is significantly non-zero under significance level 0.05. Therefore, under the assumption that smoking has uniform effect on death rate across all age groups, and adjusting for age and squared age removes confounding, we conclude smoking increases death rate (This statement is actually still problematic in a causal inference perspective, but for now we will leave it as is). As in logistic regression, often we are more interested in rate ratios rather than log rate ratios, so we may produce the rate ratio estimate and confidence interval with:

```
exp(coef(pois_mod))

##  (Intercept)        smoke          age     I(age^2)
## 1.738783e-08 1.425497e+00 1.385552e+00 9.980581e-01

exp(confint.default(pois_mod))

##                    2.5 %       97.5 %
## (Intercept) 2.183142e-09 1.384869e-07
## smoke       1.154970e+00 1.759389e+00
## age         1.295568e+00 1.481786e+00
## I(age^2)    9.975271e-01 9.985894e-01
```

So we have the rate ratio estimate for smoking as `1.425` with confidence interval `(1.155 - 1.759)`, which does not include 1 and fits our conclusion in the Wald test. We can also produce the likelihood ratio-based interval, which will be slightly different from the Wald-based interval:

```
exp(confint(pois_mod))

## Waiting for profiling to be done...

##                    2.5 %       97.5 %
## (Intercept) 2.087455e-09 1.327550e-07
## smoke       1.160865e+00 1.769171e+00
## age         1.297035e+00 1.483580e+00
## I(age^2)    9.975185e-01 9.985814e-01
```

Now we turn to the second question: Is there differential effect of smoking related to age? From the plot above, its seems that the smoking increases the death rate in the age range of $35-74$ years old, where for those of $75-84$ years old, smoking is associated with lower death rate. However, we are not sure if this discrepancy reflect true interaction or is just a result of random variation. To test for existence of interaction, We can fit the following model with interaction terms (here we do not want to impose any assumptions on the pattern of interaction, so we used categorical age groups for the interaction):

```
pois_mod_inter <- glm(r ~ offset(log(n)) + smoke + age + I(age^2) + smoke:factor(age),
                      family = poisson(), data = data)
summary(pois_mod_inter)

##
## Call:
## glm(formula = r ~ offset(log(n)) + smoke + age + I(age^2) + smoke:factor(age),
##     family = poisson(), data = data)
##
## Deviance Residuals:
##       1        2        3        4        5        6        7        8
##  0.0000   0.0000   0.0000   0.0000   0.0000  -0.4876   0.2559   0.3937
##       9       10
## -0.6487   0.2643
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -2.149e+01  3.200e+00  -6.716 1.86e-11 ***
## smoke                1.421e+00  4.938e-01   2.877  0.00401 **
## age                  4.136e-01  1.005e-01   4.115 3.88e-05 ***
## I(age^2)            -2.421e-03  7.745e-04  -3.126  0.00177 **
## smoke:factor(age)50 -5.858e-01  3.764e-01  -1.556  0.11963
## smoke:factor(age)60 -9.614e-01  5.244e-01  -1.833  0.06676 .
## smoke:factor(age)70 -1.236e+00  5.591e-01  -2.211  0.02701 *
## smoke:factor(age)80 -1.473e+00  5.085e-01  -2.897  0.00377 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 935.06733  on 9  degrees of freedom
## Residual deviance:   0.94896  on 2  degrees of freedom
## AIC: 72.017
##
## Number of Fisher Scoring iterations: 4
```

Note that four more variables representing interaction is adding into the model by `smoke:factor(age)`. The explanation for the regression coefficients corresponding to the interaction terms are more involved. Under this model, for subjects within the 40-year-old age group, the effect of smoking on death rate is represented by the log odds ratio estimate `1.421` (since for this age group, all interaction terms are zero). For subjects within the 50-year-old age group, the effect of smoking is instead `1.421-0.5858`. This can be derived by picturing a subject of age within $45-54$ years old, say, 50 years old. If this subject did not smoke, their log event rate would be

$$\beta_{(\text{Intercept})} + 50\beta_{\text{age}} + 2500\beta_{\text{age}^2} \tag{176}$$

If this subject smoked, their log event rate would then be

$$\beta_{(\text{Intercept})} + \beta_{\text{smoke}} + 50\beta_{\text{age}} + 2500\beta_{\text{age}^2} + \beta_{\text{smoke:(age = 50)}} \tag{177}$$

Therefore, the log rate ratio between smoking and not smoking would be $\beta_{\text{smoke}} + \beta_{\text{smoke:(age = 50)}}$. Consequently, the regression coefficient for interaction term `smoke:factor(age)50` is the *difference of log rate ratio for smoking between the* 50*-year-old group and the* 40*-year-old group*. Alternatively, it is the *logarithm of the ratio of rate ratio for smoking between the* 50*-year-old group and the* 40*-year-old group*. Therefore, to test for the null hypothesis that there is no interaction, i.e. the effect

of smoking is uniform across all age groups, we should test if all regression coefficients for the interaction terms are simultaneously zero. For this, the easiest way is to use likelihood ratio test (LRT), since the original model is nested within the interaction model with all coefficients for the interaction terms set to zero.

```
anova(pois_mod, pois_mod_inter, test = "LRT")

## Analysis of Deviance Table
##
## Model 1: r ~ offset(log(n)) + smoke + age + I(age^2)
## Model 2: r ~ offset(log(n)) + smoke + age + I(age^2) + smoke:factor(age)
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1         6     12.175
## 2         2      0.949  4   11.227  0.02413 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The test rejects the null hypothesis under significance level of 0.05, which indicates that there *is* interaction between age group and smoking with respect to death rate, i.e. smoking has differential effect among different age groups. So to correctly model the data, we should use the model with interaction terms. However, as we have elaborated, the regression coefficients for this model cannot be easily explain. We can therefore fit an equivalent model and calculate the estimate and confidence intervals for the exponential of its regression coefficients:

```
pois_mod_inter_2 <- glm(r ~ offset(log(n)) + age + I(age^2) + smoke:factor(age),
                        family = poisson(), data = data)
summary(pois_mod_inter_2)

##
## Call:
## glm(formula = r ~ offset(log(n)) + age + I(age^2) + smoke:factor(age),
##     family = poisson(), data = data)
##
## Deviance Residuals:
##       1        2        3        4        5        6        7        8
##  0.0000   0.0000   0.0000   0.0000   0.0000  -0.4876   0.2559   0.3937
##       9       10
## -0.6487   0.2643
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -2.149e+01  3.200e+00  -6.716 1.86e-11 ***
## age                  4.136e-01  1.005e-01   4.115 3.88e-05 ***
## I(age^2)            -2.421e-03  7.745e-04  -3.126 0.001770 **
## smoke:factor(age)40  1.421e+00  4.938e-01   2.877 0.004009 **
## smoke:factor(age)50  8.350e-01  2.232e-01   3.741 0.000183 ***
## smoke:factor(age)60  4.594e-01  1.601e-01   2.869 0.004117 **
## smoke:factor(age)70  1.844e-01  1.459e-01   1.264 0.206228
## smoke:factor(age)80 -5.227e-02  2.024e-01  -0.258 0.796235
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 935.06733  on 9  degrees of freedom
## Residual deviance:   0.94896  on 2  degrees of freedom
## AIC: 72.017
##
## Number of Fisher Scoring iterations: 4

exp(coef(pois_mod_inter_2))
```

```
##         (Intercept)                 age            I(age^2) smoke:factor(age)40
##        4.643084e-10        1.512203e+00        9.975816e-01        4.140407e+00
## smoke:factor(age)50 smoke:factor(age)60 smoke:factor(age)70 smoke:factor(age)80
##        2.304902e+00        1.583140e+00        1.202552e+00        9.490747e-01
```

```
exp(confint(pois_mod_inter_2))
```

```
## Waiting for profiling to be done...
```

```
##                              2.5 %       97.5 %
## (Intercept)         5.606434e-13 1.673345e-07
## age                 1.253952e+00 1.862964e+00
## I(age^2)            9.959920e-01 9.990337e-01
## smoke:factor(age)40 1.689421e+00 1.188389e+01
## smoke:factor(age)50 1.520841e+00 3.673272e+00
## smoke:factor(age)60 1.165776e+00 2.186095e+00
## smoke:factor(age)70 9.069869e-01 1.608114e+00
## smoke:factor(age)80 6.456479e-01 1.430959e+00
```

Here although we took out the `smoke` term, the `smoke:factor(age)40` is added back, so this model is actually equivalent to the previous interaction model. Now the regression coefficient for the interaction terms `smoke:factor(age)40` is still the log rate ratio for smoking within the 40-year-old age group, but the regression coefficient for `smoke:factor(age)50` is the log rate ratio for smoking within the 50-year-old age group (why?). Therefore, the exponentiated coefficients can be directly interpreted as the rate ratios for smoking within each age group.

8.2 Overdispersion in Generalized Linear Models

In the second handout when we talked about deviance, we derived that a Poisson model (with number of covariate patterns equal to the sample size) would have deviance

$$D = 2\sum_i \boldsymbol{Y}_i \log\left(\frac{\boldsymbol{Y}_i}{\hat{\lambda}_i}\right) \tag{178}$$

And in the third handout we showed the following Taylor expansion approximation:

$$x\log\left(\frac{x}{x_0}\right) \approx (x - x_0) + \frac{1}{2}\frac{(x-x_0)^2}{x_0} \tag{179}$$

Therefore, given $\boldsymbol{Y}_i$ is close to $\hat{\lambda}_i$, the estimated mean number of events for observation i, we have

$$D = 2\sum_i \boldsymbol{Y}_i \log\left(\frac{\boldsymbol{Y}_i}{\hat{\lambda}_i}\right) \approx 2\sum_i \left[(\boldsymbol{Y}_i - \hat{\lambda}_i) + \frac{1}{2}\frac{(\boldsymbol{Y}_i - \hat{\lambda}_i)^2}{\hat{\lambda}_i}\right] \tag{180}$$

Further using the fact that in GLMs with intercepts, the grand mean is well calibrated (we have also used this in the second handout when deriving the deviance), we have $\frac{1}{n}\sum_i \boldsymbol{Y}_i = \frac{1}{n}\sum_i \hat{\lambda}_i$, or $\sum_i(\boldsymbol{Y}_i - \hat{\lambda}_i) = 0$. Therefore

$$D \approx 2\sum_i \left[(\boldsymbol{Y}_i - \hat{\lambda}_i) + \frac{1}{2}\frac{(\boldsymbol{Y}_i - \hat{\lambda}_i)^2}{\hat{\lambda}_i}\right] = \sum_i \frac{(\boldsymbol{Y}_i - \hat{\lambda}_i)^2}{\hat{\lambda}_i} := X^2 \tag{181}$$

X^2 is the Pearson chi-squared statistic for Poisson regression models, which should also be asymptotically distributed as $\chi^2_{(n-p)}$ when the null hypothesis of well-fitted model is true. Inspecting the X^2 statistic, we see that since the variance and mean of a Poisson-distributed random variable are both λ, the statistic is actually of the form:

$$X^2 = \sum_i \frac{(o_i - e_i)^2}{\hat{var}(o_i)} \tag{182}$$

where o_i and e_i stand for the observed and expected outcome of observation i. An ill-fitted model would have a large X^2, which means that actually variance of o_i, reflected by $(o_i - e_i)^2$ is actually much larger than the estimated variance of o_i. That is, the variance, or *dispersion* of the observed

variance is larger than expected by the model. Therefore, if a model is ill-fitted reflected by a X^2 (or D) much larger than $n-p$, we say that there is *overdispersion*. In practice, we can treat overdispersion as a synonym for "badness of fit" in GLMs.

A simple (but not very insightful) way to deal with overdispersion is to plug in a larger dispersion parameter ϕ, which we have always assumed to be 1 in Poisson regression. Specifically, since the null distribution of D (or X^2) should be $\chi^2_{(n-p)}$ that has a mean of $n-p$, it is common to just set $\phi = D/(n-p)$ (or $X^2/(n-p)$), so that the new deviance (or Pearson chi-squared statistic) would be just $n-p$. Setting the dispersion parameter to be larger than 1 would lead to the same MLE estimates for the regression coefficients, but a larger confidence interval. This is intuitive since the (single-observation) log-likelihood of the one-parameter canonical exponential family with dispersion is (setting $a(\phi) = \phi$):

$$\ell_i(\theta_i) = \frac{y_i\theta_i - b(\theta_i)}{\phi} + c(y_i, \phi) \tag{183}$$

For the MLE, $c(y, \phi)$ does not influence the estimate since we will be differentiating ℓ by θ, the score would then be

$$U(\beta) = \sum_i \frac{\partial}{\partial\beta}\ell_i(\theta_i) = \frac{1}{\phi}\sum_i (y_i - b'(\theta_i))\frac{\partial\theta_i}{\partial\beta} \tag{184}$$

Therefore, ϕ would not influence the MLE estimate since we are just solving $U(\beta) = 0$. However, the estimated variance of $\hat{\beta}$ would become larger with $\phi > 1$ since now the Fisher information would be multiplied by a factor of $\frac{1}{\phi}$, so the estimated variance-covariance matrix for the MLE, which is the inverse of the Fisher information, would be multiplied by a factor of ϕ. Let us fit a very bad model on our previous data to show this:

```
pois_mod_od <- glm(r ~ offset(log(n)) + age, family = poisson(), data = data)
(nodisp <- summary(pois_mod_od))

##
## Call:
## glm(formula = r ~ offset(log(n)) + age, family = poisson(), data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.8277  -2.7448  -0.9911   0.5406   4.5992
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -10.298792   0.188727  -54.57   <2e-16 ***
## age           0.083776   0.002889   28.99   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 935.067  on 9  degrees of freedom
## Residual deviance:  85.012  on 8  degrees of freedom
## AIC: 144.08
##
## Number of Fisher Scoring iterations: 4

(phi <- pois_mod_od$deviance / pois_mod_od$df.residual)

## [1] 10.62644

(disp <- summary(pois_mod_od, dispersion = phi))

##
## Call:
## glm(formula = r ~ offset(log(n)) + age, family = poisson(), data = data)
##
```

```
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.8277  -2.7448  -0.9911   0.5406   4.5992
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -10.298792   0.615217 -16.740   <2e-16 ***
## age           0.083776   0.009419   8.894   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 10.62644)
##
##     Null deviance: 935.067  on 9  degrees of freedom
## Residual deviance:  85.012  on 8  degrees of freedom
## AIC: 144.08
##
## Number of Fisher Scoring iterations: 4

(disp$coefficients[,2] / nodisp$coefficients[,2])^2

## (Intercept)         age
##    10.62644    10.62644
```

Here we can see the in the original "bad" model with only `age` as explanatory variable, the residual deviance is `85.012`, which is 10 times its degrees of freedom $8 = 10 - 2$. If we set the dispersion parameter as `85.012/8 = 10.626`, then we can see that the parameter estimates remains the same, but the standard error of the estimates becomes larger. In fact, we can verify that the squared ratio between the standard errors in these two models is exactly the dispersion parameter, just as we would expect from the theoretical arguments above.

The reason why we said that using estimated dispersion parameters is not insightful is that it only addresses the problem by inflating the variance of the parameter estimates, but does not look into *why* there is overdispersion, or badness of fit. Common reasons for overdispersion are lists as follows, with some "more insightful" strategies to deal with them:

- Ill-specified variable functional form: Transform the variables and fit again
- Unmeasured explanatory variables: Try to gather more explanatory variables
- Individual-specific effects: Fit a generalized linear mixed model (GLMM), generalized estimating equation (GEE) or negative binomial model (beta-binomial model for binary outcome GLMs)
- Correlated observations: Fit a GLMM or GEE

In the above, the negative binomial model assumes that the event rate for each individual actually comes from a Gamma-distributed distribution with mean $\exp(X\beta)$, and the beta-binomial model assumes that the event probability for each individual comes from a Beta distribution with mean $\text{antilogit}(X\beta)$. The details of these models is out of scope for this course, so we will stop here.