Chapter 4: Statistical Models and their Inferential Properties

In this chapter, we will explore how the methods of statistical inference that we developed for the setting of n IID observations $X_1, \ldots, X_n \overset{IID}{\sim} f(x \mid \theta)$ may be applied to other types of data and statistical models. We will introduce several statistical models using different motivating examples and then use our tools from previous chapters to solve different inferential questions[1]:

1. Linear Regression
2. Logistic Regression
3. Poisson Regression
4. Cox Proportional Hazards (CoxPH) Model
5. Bradley Terry Model

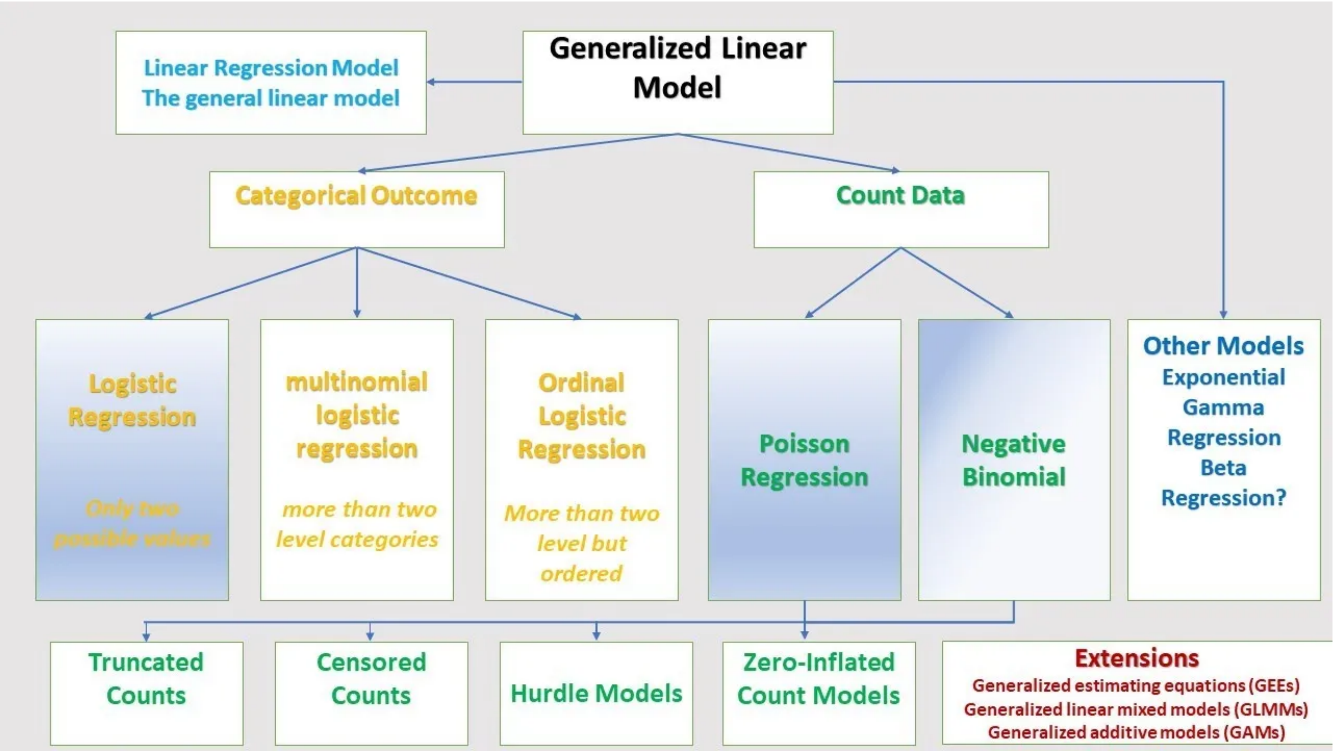


Figure 1: GLM Family of Models (Pic from YouTube Channel @UCLAOARC)

A parametric model for a data vector $\mathbf{Y}$ (not necessarily consisting of IID coordinates) is a specification of the joint distribution of $\mathbf{Y}$ in terms of a small number of parameters θ. The likelihood $\operatorname{lik}(\theta) = f(\mathbf{Y} \mid \theta)$ is the joint PMF or PDF of $\mathbf{Y}$ viewed as a function of θ. The log-likelihood is $l(\theta) = \log \operatorname{lik}(\theta)$, and the MLE $\hat{\theta}$ is the value of θ that maximizes $\operatorname{lik}(\theta)$. To extend the theory of maximum likelihood and Fisher

[1] Key References: 1. Categorical Data Analysis Book; 2. ISLR Book; 3. FPP Book

information to the non-IID setting, note that for IID data $X_1, \ldots, X_n \overset{IID}{\sim} f(x \mid \theta)$, we may introduce the notation

$$I_{\mathbf{X}}(\theta) := nI(\theta) = \sum_{i=1}^{n} -\mathbb{E}_\theta \left[\frac{\partial^2}{\partial \theta^2} \log f\left(X_i \mid \theta\right)\right] = -\mathbb{E}_\theta \left[l''(\theta)\right],$$

which represents the total Fisher information of all n observations $\mathbf{X} = (X_1, \ldots, X_n)$. Our main theorem regarding the MLE $\hat{\theta}$ states that it is approximately distributed as $\mathcal{N}\left(\theta_0, \frac{1}{n} I\left(\theta_0\right)^{-1}\right) = \mathcal{N}\left(\theta_0, I_{\mathbf{X}}\left(\theta_0\right)^{-1}\right)$ for large n if the parametric model is correct and the true parameter is θ_0. For non-IID data and the general log-likelihood $l(\theta) = \log f(\mathbf{Y} \mid \theta)$, let us define $I_{\mathbf{Y}}(\theta) = -\mathbb{E}_\theta \left[l''(\theta)\right]$ in the single-parameter case $\theta \in \mathbb{R}$ and $I_{\mathbf{Y}}(\theta) = -\mathbb{E}_\theta \left[\nabla^2 l(\theta)\right]$ in the multi-parameter case $\theta \in \mathbb{R}^k$, where

$$\nabla^2 l(\theta) = \left(\frac{\partial^2}{\partial \theta_i \partial \theta_j} l(\theta)\right)_{1 \leq i,j \leq k}$$

is the second-derivative (Hessian) matrix for $l(\theta)$. In all of the non-IID settings we will consider, under appropriate asymptotic conditions, the approximate sampling distribution of $\hat{\theta}$ is still given by the (multivariate) normal distribution $\mathcal{N}\left(\theta_0, I_{\mathbf{Y}}\left(\theta_0\right)^{-1}\right)$ when the total sample size is large.

1 The Linear Model

Example 1.0.1. *When a string instrument sustains a note at a particular pitch, the resulting sound wave is periodic with some fixed frequency f (say 440 Hz). For a "pure" tone at this pitch, the sound wave is a perfect sinusoidal wave with frequency f, but the sound produced by any real string instrument is not a pure tone. Instead, it is a superposition of sinusoidal waves with frequencies $f, 2f, 3f, 4f$, etc., corresponding to different vibrating modes of the string. These frequencies are called resonance harmonics, and the relative volumes, or amplitudes, of the resonance harmonics, determine the timbre or "color" of the sound.*

A recording device measures the sound wave produced by an instrument (sustaining a single note) at n points in time $t_1, \ldots, t_n$, where the measurements are contaminated by white noise. We consider the problem of estimating the amplitudes of the resonance harmonics for this instrument. Let $Y_1, \ldots, Y_n \in \mathbb{R}$ be measurements of the sound wave at these time points. Suppose, for simplicity, we have scaled our units so that the sine and cosine curves corresponding to the fundamental frequency f are $\sin(t)$ *and* $\cos(t)$. *Then, assuming the existence of resonance harmonics up to frequency kf, we may model each measurement Y_i as*

$$Y_i = \beta_1 \sin\left(t_i\right) + \beta_2 \cos\left(t_i\right) + \beta_3 \sin\left(2t_i\right) + \beta_4 \sin\left(2t_i\right) + \ldots + \beta_{2k-1} \sin\left(kt_i\right) + \beta_{2k} \cos\left(kt_i\right) + \varepsilon_i, \qquad (1)$$

for some coefficients $\beta_1, \ldots, \beta_{2k} \in \mathbb{R}$, where the errors $\varepsilon_i \overset{IID}{\sim} \mathcal{N}\left(0, \sigma_0^2\right)$ correspond to the white noise and the variance σ_0^2 signifies the noise level. If we construct the matrix

$$X = \begin{pmatrix} \sin\left(t_1\right) & \cos\left(t_1\right) & \cdots & \sin\left(kt_1\right) & \cos\left(kt_1\right) \\ \sin\left(t_2\right) & \cos\left(t_2\right) & \cdots & \sin\left(kt_2\right) & \cos\left(kt_2\right) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \sin\left(t_n\right) & \cos\left(t_n\right) & \cdots & \sin\left(kt_n\right) & \cos\left(kt_n\right) \end{pmatrix}$$

and denote its entries as x_{ij}, then we may write the above as

$$Y_i = \sum_{j=1}^{2k} \beta_j x_{ij} + \varepsilon_i,$$

or more succinctly in matrix notation, for all $i = 1, \ldots, n$, as

$$Y = X\beta + \varepsilon.$$

*Here, Y denotes the column vector $(Y_1, \ldots, Y_n)$, β denotes the column vector $(\beta_1, \ldots, \beta_{2k})$, and ε denotes the column vector $(\varepsilon_1, \ldots, \varepsilon_n)$. This is called a **linear model**.*

More generally, given a vector of **responses** $Y_1, \ldots, Y_n$, the linear model models each Y_i as a certain linear combination $\beta_1 x_{i1} + \ldots + \beta_p x_{ip}$ of corresponding **covariates** $x_{i1}, \ldots, x_{ip}$, plus IID Gaussian errors. (The coefficients $\beta_1, \ldots, \beta_p$ are the same for all n responses $Y_1, \ldots, Y_n$.) We will treat the covariates as fixed and known constants. The values of the Gaussian errors are not directly observed; for simplicity, however, we'll assume in this section that their variance σ_0^2 is known. The parameters of the model are the regression coefficients $\beta_1, \ldots, \beta_p$. (Much of our analysis in this section may be extended to the more realistic setting where σ_0^2 is unknown, in which case it would also be a parameter of the model.)

1.1 Statistical inference

In the model of equation (1), the amplitudes of the k resonance harmonics are defined as $A_1 = \sqrt{\beta_1^2 + \beta_2^2}, A_2 = \sqrt{\beta_3^2 + \beta_4^2}, \ldots, A_k = \sqrt{\beta_{2k-1}^2 + \beta_{2k}^2}$. We will discuss the following inferential tasks:

- Estimate the amplitudes $A_1, \ldots, A_k$
- Provide confidence intervals corresponding to these estimates

Let $p = 2k$. To write down the likelihood for the linear model, note that $Y_1, \ldots, Y_n$ are independent and distributed as $Y_i \sim \mathcal{N}\left(\sum_j \beta_j x_{ij}, \sigma_0^2\right)$. Then

$$\text{lik}\left(\beta_1, \ldots, \beta_p\right) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma_0^2}} \exp\left(-\frac{1}{2\sigma_0^2}\left(Y_i - \sum_{j=1}^{p} \beta_j x_{ij}\right)^2\right),$$

and the log-likelihood is

$$l\left(\beta_1, \ldots, \beta_p\right) = -\frac{n}{2}\log\left(2\pi\sigma_0^2\right) - \frac{1}{2\sigma_0^2}\sum_{i=1}^{n}\left(Y_i - \sum_{j=1}^{p} \beta_j x_{ij}\right)^2.$$

For any $\sigma_0^2 > 0$, this log-likelihood is maximized when $\beta_1, \ldots, \beta_k$ are the **least-squares estimators** minimizing the total squared error

$$\text{err} = \sum_{i=1}^{n}\left(Y_i - \sum_{j=1}^{p} \beta_j x_{ij}\right)^2.$$

So the MLEs $\hat{\beta}_1, \ldots, \hat{\beta}_k$ are equal to the least-squares estimators. To compute these MLEs, we solve the system of p equations for $m = 1, \ldots, p$

$$0 = \frac{\partial l}{\partial \beta_m} = \frac{1}{\sigma_0^2}\sum_{i=1}^{n} x_{im}\left(Y_i - \sum_{j=1}^{p} \beta_j x_{ij}\right).$$

Letting X_m denote the m th column of X, these equations may be written as

$$0 = \frac{1}{\sigma_0^2} X_m^T (Y - X\beta),$$

or even more succinctly for all $m = 1, \ldots, p$ as $0 = \frac{1}{\sigma_0^2} X^T(Y - X\beta)$, where both sides of this equation are vectors of length p. Solving for β yields the MLEs/least-squares estimates

$$\hat{\beta} = \left(X^T X\right)^{-1} X^T Y.$$

To estimate an amplitude of a resonance harmonic, say $A_1 = \sqrt{\beta_1^2 + \beta_2^2}$, we may use the plugin estimate $\hat{A}_1 = \sqrt{\hat{\beta}_1^2 + \hat{\beta}_2^2}$.

To obtain a confidence interval for A_1, we will derive an approximate standard error for $\hat{A}_1$, by first deriving the sampling distribution of $\hat{\beta}$ and then applying the delta method. We compute the Fisher information $I_{\mathbf{Y}}(\beta) = -\mathbb{E}_\beta \left[\nabla^2 l(\beta)\right]$, by computing the second-order derivatives of l:

$$\frac{\partial^2 l}{\partial \beta_m \partial \beta_l} = -\frac{1}{\sigma_0^2} \sum_{i=1}^{n} x_{im} x_{il} = -\frac{1}{\sigma_0^2} X_m^T X_l$$

Then the Hessian matrix is $\nabla^2 l(\beta) = -\frac{1}{\sigma_0^2} X^T X$, so $I_{\mathbf{Y}}(\beta) = \frac{1}{\sigma_0^2} X^T X$. The distribution of $\hat{\beta}$ is then approximately $\mathcal{N}\left(\beta, \sigma_0^2 \left(X^T X\right)^{-1}\right)$ for large n.

In fact, the distribution of $\hat{\beta}$ is exactly this multivariate normal distribution even for small n, because $Y = X\beta + \varepsilon \sim \mathcal{N}\left(X\beta, \sigma_0^2 I\right)$ so that $\hat{\beta} = \left(X^T X\right)^{-1} X^T Y \sim \mathcal{N}\left(\left(X^T X\right)^{-1} X^T X \beta, \sigma_0^2 \left(X^T X\right)^{-1} X^T X \left(X^T X\right)^{-1}\right) = \mathcal{N}\left(\beta, \sigma_0^2 \left(X^T X\right)^{-1}\right)$.

We now apply the **delta method:** Defining $g(x, y) = \sqrt{x^2 + y^2}$, a Taylor expansion yields

$$\begin{aligned} \hat{A}_1 - A_1 = g\left(\hat{\beta}_1, \hat{\beta}_2\right) - g\left(\beta_1, \beta_2\right) &\approx \frac{\partial g}{\partial x}\left(\beta_1, \beta_2\right) \times \left(\hat{\beta}_1 - \beta_1\right) + \frac{\partial g}{\partial y}\left(\beta_1, \beta_2\right) \times \left(\hat{\beta}_2 - \beta_2\right) \\ &= \frac{\beta_1}{\sqrt{\beta_1^2 + \beta_2^2}} \left(\hat{\beta}_1 - \beta_1\right) + \frac{\beta_2}{\sqrt{\beta_1^2 + \beta_2^2}} \left(\hat{\beta}_2 - \beta_2\right). \end{aligned}$$

Letting $c_1 = \beta_1 / \sqrt{\beta_1^2 + \beta_2^2}, c_2 = \beta_2 / \sqrt{\beta_1^2 + \beta_2^2}, Z_1 = \hat{\beta}_1 - \beta_1$, and $Z_2 = \hat{\beta}_2 - \beta_2$, the above sampling distribution for $\hat{\beta}$ implies that (Z_1, Z_2) is approximately bivariate normal with mean 0 and covariance given by the upper-left 2×2 block of $\sigma_0^2 \left(X^T X\right)^{-1}$. So $\hat{A}_1 - A_1 \approx c_1 Z_1 + c_2 Z_2$ is approximately normal with mean 0 and variance

$$\begin{aligned} \operatorname{Var}\left[c_1 Z_1 + c_2 Z_2\right] &= \operatorname{Cov}\left[c_1 Z_1 + c_2 Z_2, c_1 Z_1 + c_2 Z_2\right] \\ &= c_1^2 \operatorname{Var}\left[Z_1\right] + c_2^2 \operatorname{Var}\left[Z_2\right] + 2 c_1 c_2 \operatorname{Cov}\left[Z_1, Z_2\right] \\ &= c_1^2 \sigma_0^2 \left(\left(X^T X\right)^{-1}\right)_{11} + c_2^2 \sigma_0^2 \left(\left(X^T X\right)^{-1}\right)_{22} + 2 c_1 c_2 \sigma_0^2 \left(\left(X^T X\right)^{-1}\right)_{12}. \end{aligned}$$

Letting $\hat{c}_1 = \hat{\beta}_1 / \sqrt{\hat{\beta}_1^2 + \hat{\beta}_2^2}$ and $\hat{c}_2 = \hat{\beta}_2 / \sqrt{\hat{\beta}_1^2 + \hat{\beta}_2^2}$, we may estimate the standard error of $\hat{A}_1$ by

$$\hat{\mathrm{se}} = \sqrt{\hat{c}_1^2 \sigma_0^2 \left(\left(X^T X\right)^{-1}\right)_{11} + \hat{c}_2^2 \sigma_0^2 \left(\left(X^T X\right)^{-1}\right)_{22} + 2 \hat{c}_1 \hat{c}_2 \sigma_0^2 \left(\left(X^T X\right)^{-1}\right)_{12}},$$

and construct a 95% confidence interval for A_1 as $\hat{A}_1 \pm z(0.025)\hat{\mathrm{se}}$.

2 Logistic Regression

2.1 The logistic regression model

Example 2.1.1. *An internet company would like to understand what factors influence whether a visitor to a webpage clicks on an advertisement. Suppose it has available historical data of n ad impressions, each impression corresponding to a single ad being shown to a single visitor. For the i^{th} impression, let $Y_i \in \{0,1\}$ be such that $Y_i = 1$ if the visitor clicked on the ad, and $Y_i = 0$ otherwise. The internet company also has available various attributes for each impression, such as the position and size of the ad on the webpage, the company or product being advertised, the age and gender of the visitor, the time of day, the month of the year, etc. For each i^{th} impression, suppose that all of these attributes are encoded numerically as p covariates $x_{i1}, \ldots, x_{ip} \in \mathbb{R}$.*

Logistic regression model assumes each response Y_i is an independent random variable with distribution Bernoulli (p_i), where the log-odds corresponding to p_i is modeled as a linear combination of the covariates plus a possible intercept term:

$$\log \frac{p_i}{1-p_i} = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}.$$

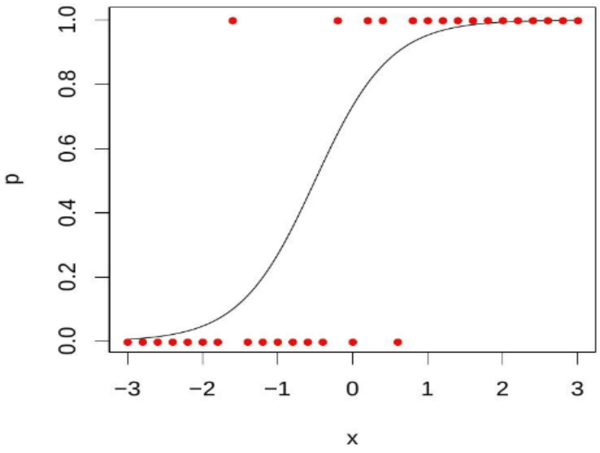
Note: Loosely speaking, p_i ranges from 0 to 1 and odds $\left(\frac{p_i}{1-p_i}\right)$ range between 0 to ∞. Therefore, Log-odds range from $-\infty$ to ∞. That is why the log odds are used to avoid modeling a variable with a restricted range, such as probability.

The intercept β_0 represents the "baseline" log-odds of the visitor clicking on the ad if all of the covariates take the value 0. Each coefficient β_j represents the amount of increase or decrease in the log-odds if the value of the covariate x_{ij} is increased by 1 unit. The above may be equivalently written as

$$\mathbb{P}\left[Y_i = 1\right] = p_i = \frac{e^{\beta_0+\beta_1 x_{i1}+\ldots+\beta_p x_{ip}}}{1+e^{\beta_0+\beta_1 x_{i1}+\ldots+\beta_p x_{ip}}}. \tag{2}$$

As in the case of the linear model, we will treat the covariates as fixed and known quantities. The unknown parameters are the regression coefficients $\beta = (\beta_0, \ldots, \beta_p)$.

When there is only one covariate, $p = 1$, we simply write $x_1 = x_{11}, \ldots, x_n = x_{n1}$. The picture below illustrates the logistic regression model, where the red points correspond to the data values $(x_1, Y_1), \ldots, (x_n, Y_n)$ of the covariate and response, and the black curve shows the probability function $p(x) = \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}}$:



2.2 Statistical inference

We will explore the following inferential questions:

- Estimate the regression coefficients $\beta_0, \beta_1, \ldots, \beta_p$
- Estimate the "conversion" probability that a new impression, with covariate values $(\tilde{x}_1, \ldots, \tilde{x}_p)$, will lead to click on the ad
- Test whether $\beta_j = 0$ for a particular covariate j, say the age of the visitor, and provide a confidence interval for β_j

Since the responses $Y_1, \ldots, Y_n$ are independent Bernoulli random variables, the likelihood for the logistic regression model is given by

$$\operatorname{lik}(\beta_0, \ldots, \beta_p) = \prod_{i=1}^{n} p_i^{Y_i} (1 - p_i)^{1-Y_i} = \prod_{i=1}^{n} (1 - p_i) \left(\frac{p_i}{1 - p_i} \right)^{Y_i},$$

where p_i is defined as a function of $\beta_0, \ldots, \beta_p$ and the covariates $x_{i1}, \ldots, x_{ip}$ by equation (2). Then, introducing for convenience a covariate $x_{i0} \equiv 1$ for all i that captures the intercept term, the log-likelihood is

$$l(\beta_0, \ldots, \beta_p) = \sum_{i=1}^{n} Y_i \log \frac{p_i}{1 - p_i} + \log(1 - p_i) = \sum_{i=1}^{n} \left(Y_i \sum_{j=0}^{p} \beta_j x_{ij} - \log\left(1 + e^{\sum_{j=0}^{p} \beta_j x_{ij}}\right) \right).$$

To estimate the parameters $\beta_0, \ldots, \beta_p$, we may compute the MLE. For the function $f(x) = \log(1 + e^x)$, $f'(x) = \frac{e^x}{1+e^x} = 1 - \frac{1}{1+e^x}$. Then, setting the partial derivatives of the log-likelihood equal to 0 and applying the chain rule yields the equations (for $m = 0, \ldots, p$)

$$0 = \frac{\partial l}{\partial \beta_m} = \sum_{i=1}^{n} x_{im} \left(Y_i - \frac{e^{\sum_{j=0}^{p} \beta_j x_{ij}}}{1 + e^{\sum_{j=0}^{p} \beta_j x_{ij}}} \right). \tag{3}$$

These equations may be solved numerically (e.g., by Newton-Raphson) to obtain the MLEs $\hat{\beta}_0, \ldots, \hat{\beta}_p$. To estimate the conversion probability for a new impression with covariates $\tilde{x}_1, \ldots, \tilde{x}_p$, we may use the plugin estimate.

$$\hat{p} = \frac{e^{\hat{\beta}_0 + \hat{\beta}_1 \tilde{x}_1 + \ldots + \hat{\beta}_p \tilde{x}_p}}{1 + e^{\hat{\beta}_0 + \hat{\beta}_1 \tilde{x}_1 + \ldots + \hat{\beta}_p \tilde{x}_p}}.$$

To test if a particular coefficient is 0, say $H_0 : \beta_p = 0$, one method uses the generalized likelihood ratio test. This null hypothesis corresponds to a sub-model with one fewer free parameters. We may calculate the sub-model MLEs $\hat{\beta}_{0,0}, \ldots, \hat{\beta}_{0,p-1}$ from the same score equations as (3) except with the p^{th} covariate removed, and use the generalized likelihood ratio statistic

$$-2 \log \Lambda = -2 \log \frac{\operatorname{lik}\left(\hat{\beta}_{0,0}, \ldots, \hat{\beta}_{0,p-1}, 0\right)}{\operatorname{lik}\left(\hat{\beta}_0, \ldots, \hat{\beta}_p\right)}.$$

When the number of impressions n is large, we may perform an approximate level-α test of H_0 by rejecting H_0 when $D > \chi_1^2(\alpha)$, since the difference between model dimensionalities here is 1.

We may obtain a confidence interval for β_j from the MLE estimate $\hat{\beta}_j$ and an estimate of its standard error: We compute the Fisher information $I_{\mathbf{Y}}(\beta) = -\mathbb{E}_\beta\left[\nabla^2 l(\beta)\right]$ by calculating the second partial derivatives of l: For $f(x) = \log\left(1+e^x\right), f''(x) = \frac{e^x}{(1+e^x)^2}$. Then

$$\frac{\partial^2 l}{\partial\beta_m \partial\beta_l} = -\sum_{i=1}^n x_{im}x_{il}\frac{e^{\sum_{j=0}^p \beta_j x_{ij}}}{\left(1+e^{\sum_{j=0}^p \beta_j x_{ij}}\right)^2} = -X_m^T W X_l,$$

where we have set $X_j = (x_{1j}, \ldots, x_{nj})$ as the j th column of the matrix of covariates as in Section 1, and defined the $n \times n$ diagonal matrix

$$W := W(\beta) = \operatorname{diag}\left(\frac{e^{\sum_{j=0}^p \beta_j x_{1j}}}{\left(1+e^{\sum_{j=0}^p \beta_j x_{1j}}\right)^2}, \cdots, \frac{e^{\sum_{j=0}^p \beta_j x_{nj}}}{\left(1+e^{\sum_{j=0}^p \beta_j x_{nj}}\right)^2}\right).$$

So $\nabla^2 l(\beta) = -X^T W X, I_{\mathbf{Y}}(\beta) = X^T W X$, and the approximate sampling distribution of $\hat{\beta}$ for large n is $\mathcal{N}\left(\beta, \left(X^T W X\right)^{-1}\right)$. Letting $\hat{W} = W(\hat{\beta})$ be the plugin estimate of the diagonal matrix W, we may estimate the standard error of $\hat{\beta}_j$ by $\hat{\text{se}}_j = \sqrt{\left(\left(X^T \hat{W} X\right)^{-1}\right)_{jj}}$, and obtain a 95% confidence interval for β_j as $\hat{\beta}_j \pm z(0.025)\hat{\text{se}}_j$.

Remark 2.1. *A word of caution regarding model misspecification: The above standard error estimates* $\hat{\text{se}}_j$ *(which are the standard errors reported by most logistic regression software) are only expected to be accurate when the logistic regression model is correctly specified that is, when the Y_i's are truly independent random variables with distribution* $\text{Bernoulli}\,(p_i)$*, where the log-odds for each p_i is the same linear combination of the covariates. This is because, as in the case of n IID observations, the covariance of $\hat{\beta}$ is given by the inverse Fisher information only in a correctly specified model.*

Logistic regression is still frequently used as a tool for binary classification problems, even if the model does not yield an extremely accurate fit to the data, as long as the model has good classification accuracy. In such settings, the MLE $\hat{\beta}$ represents the "closest" logistic regression model (in the given covariates) to the true distribution of $Y_1, \ldots, Y_n$, in the sense of KL-divergence as in Section 3 of Chapter 3. The standard error for $\hat{\beta}_j$ may be robustly estimated using either a sandwich estimator or the non-parametric bootstrap. For the logistic regression model, the sandwich estimate of the covariance matrix of $\hat{\beta}$ is given by:

$$\left(X^T \hat{W} X\right)^{-1}\left(X^T \tilde{W} X\right)\left(X^T \hat{W} X\right)^{-1},$$

where $\tilde{W} = \operatorname{diag}\left((Y_1-\hat{p}_1)^2, \ldots, (Y_n-\hat{p}_n)^2\right)$ *and $\hat{p}_i$ is the fit probability for the i^{th} observation, defined by the right side of equation (2) with $\hat{\beta}$ in place of β. The (j,j) element of this matrix gives a sandwich estimate for the variance of $\hat{\beta}_j$.*

Alternatively, one may use the ***pairs bootstrap****, which pairs the covariates and response for each i^{th} observation into a single data vector $(x_{i1}, \ldots, x_{ip}, Y_i)$, and then draws bootstrap samples by randomly selecting, with replacement, n of these vectors. The logistic regression model is fit to each such bootstrap sample to yield an MLE $\hat{\beta}^*$, and the standard error of $\hat{\beta}_j$ is estimated by the empirical standard deviation of $\hat{\beta}_j^*$ across bootstrap samples.*

Hands-on Session. (Analysis of the Dosage Mortality Curve)

An experiment was conducted to evaluate the toxicity of gaseous carbon disulfide on flour beetle. Table 1 shows the numbers of beetles n_i that were exposed to gaseous carbon disulfide at concentrations x_i (Dose, expressed in $\log_{10}$ mg/L), as well as the numbers of beetles y_i dead after five hours of exposure. The data comes from an article by Bliss (1935), Thus, the responses y_i are binomial realizations with n_i trials at concentrations x_i. This kind of experiment can be modeled using a logistic regression model.

Dose x_i	n_i	y_i
1.6907	59	6
1.7242	60	13
1.7552	62	18
1.7842	56	28
1.8113	63	52
1.8369	59	53
1.8610	62	61
1.8839	60	60

Table 1: Beetle mortality data.

This real data example illustrates the method of maximum likelihood estimation for the parameters of the logistic regression model. Write R code to compare the results using the following methods:

(a) The Newton's method.

(b) A direct implementation. Implementing the log-likelihood function and maximizing it using the command optim(). Check if using the command glm() produces the same result or not.

(c) Also, provide a visualization of the fitted dose-response model.

Solution.

(a) We need some libraries and first to write the data (Reference: Rpubs: F. Javier Rubio).

```
# Delete Memory
rm(list=ls())
# Required packages
library(knitr)
library(plotly)
# Beetle data
dat =  cbind(c(1.6907, 1.7242, 1.7552, 1.7842, 1.8113,1.8369, 1.861, 1.8839),
             c(59, 60, 62, 56, 63, 59, 62, 60),
             c(6, 13, 18, 28, 52, 53, 61, 60))
# Naming the columns
colnames(dat) <- c("Dose", "N","x")
# Displaying the data
kable(dat, digits = 4)

ns = length(dat[,1]) # number of responses
```

```r
ni <- dat[,2] # numbers of beetles
yi <- dat[,3] # numbers of dead beetles after exposure
xi <- dat[,1] # Dose

####################################################
# Maximum Likelihood Estimation using three methods
####################################################

#--------------------------
# Using  Newton's method
#--------------------------
theta = c(0,0) # Initial value
N = 100 # Number of iterations
iter <- matrix(0,nrow=N,ncol=2)

# Running the iterations
for(i in 1:N){
  iter[i,] <- theta
  pi <- exp(theta[1] + theta[2]*xi)/( 1 + exp(theta[1] + theta[2]*xi) )
  S <- c(sum(yi - ni*pi), sum(xi*(yi - ni*pi)))
  I <- rbind(c(sum(ni*pi*(1-pi)), sum(ni*xi*pi*(1-pi))),
             c(sum(ni*xi*pi*(1-pi)), sum(ni*xi^2*pi*(1-pi))))
  theta <- theta + solve(I)%*%S

}

# Illustrating the iterations
plot(iter[1:10,], type = "p")
head(iter)

df <- data.frame(x = iter[,1], y = iter[,2], f = iter[,2])
p <- df %>%
  plot_ly(
    x = ~x,
    y = ~y,
    frame = ~f,
    type = 'scatter',
    mode = 'markers',
    showlegend = F,
    size = 40
  )
p

# MLE after 100 iterations
theta

# LD50
-theta[1]/theta[2]
```

(b) Now, we will do the direct implementation.

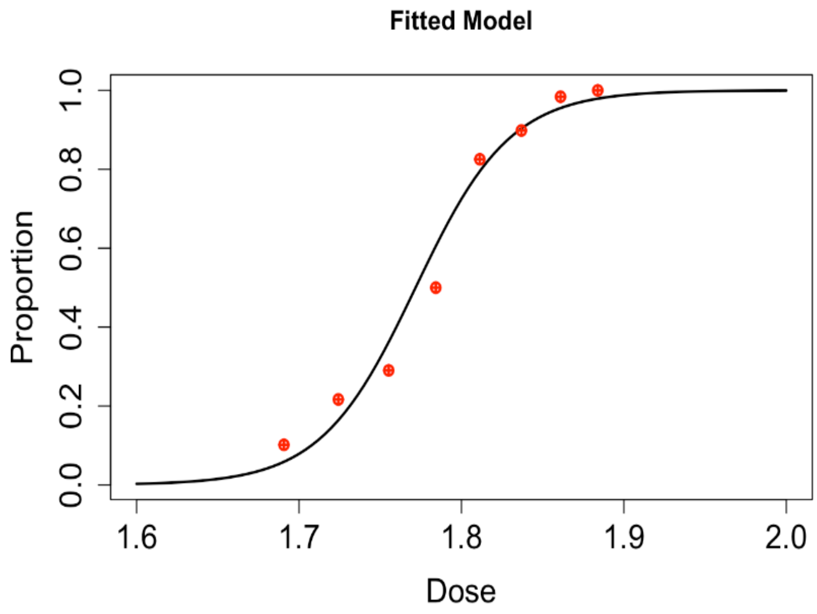
```
# log-likelihood function
lpl = function(par){
  var <- vector()
  for(i in 1:ns) var[i] = dat[i,3]*log(plogis(par[1]+par[2]*dat[i,1]))
    + (dat[i,2]-dat[i,3])*log(1-plogis(par[1]+par[2]*dat[i,1]))
  return(-sum(var))
}

# Optimisation step using optim
OPT <- optim(c(-60,30),lpl,control=list(maxit=20000,abstol=1e-18,reltol=1e-18))
# MLE
OPT$par

#---------------------------
# Using glm
#---------------------------
glm(formula = cbind(yi,ni-yi) ~ xi, family=binomial(logit), data=data.frame(dat))


## Coefficients:
## (Intercept)           xi
##      -60.72        34.27
##
## Degrees of Freedom: 7 Total (i.e. Null);  6 Residual
## Null Deviance:       284.2
## Residual Deviance: 11.23     AIC: 41.43
```

The results are virtually the same, as expected.

(c)

```
####################################################
# Visualisation of the fitted dose-response model
####################################################

# Proportions
propi <- yi/ni

# Fitted logistic model
fit.logis <- Vectorize(function(d)  plogis(theta[1] + theta[2]*d))

curve(fit.logis,1.6,2, xlab = "Dose", ylab = "Proportion",
      main = "Fitted Model", lwd = 2, cex.axis = 1.5, cex.lab = 1.5)
points(xi,propi,pch=10,col="red",lwd=2)
```

A visualization of the fitted dose-response model is as follows.



Homework. (Analysis of the NASA Challenger Disaster Data)

The NASA space shuttle ”Challenger” exploded shortly after its launch on 28 January 1986, with a loss of seven lives. The US Presidential Commission concluded that the accident was caused by a leakage of gas from one of the fuel tanks. Rubber insulating rings, so-called ”O-rings”, were not pliable enough after the overnight low temperature of 31°F, and did not plug the joint between the fuel in the tanks and the intense heat outside. Table 2 presents the data concerning previous flights, which has been slightly modified for illustration purposes by only reporting the presence of failure of the O-rings. The last row corresponds to the conditions at which the Challenger was launched.

The description of this data can be found at: Challenger Explosion Data. A brief copy-paste description is as follows:

The NASA space shuttle Challenger exploded on January 28, 1986, just 73 seconds after liftoff, bringing a devastating end to the spacecraft's 10th mission. The disaster claimed the lives of all seven astronauts aboard, including Christa McAuliffe, a teacher from New Hampshire who would have been the first civilian in space. It was later determined that two rubber O-rings, which had been designed to separate the sections of the rocket booster, had failed due to cold temperatures on the morning of the launch. The tragedy and its aftermath received extensive media coverage and prompted NASA to temporarily suspend all shuttle missions.

	Failure	Temperature	Pressure (psi)
1	0	66	50
2	1	70	50
3	0	69	50
4	0	68	50
5	0	67	50
6	0	72	50
7	0	73	100
8	0	70	100
9	1	57	200
10	1	63	200
11	1	70	200
12	0	78	200
13	0	67	200
14	1	53	200
15	0	67	200
16	0	75	200
17	0	70	200
18	0	81	200
19	0	76	200
20	0	79	200
21	1	75	200
22	0	76	200
23	1	58	200
C	-	31	200

Table 2: Challenger data. Failures out of 6 rings.

(a) Fit a logistic regression model to the data and find the values of the coefficients using MLE.

(b) Find out the Confidence Intervals (CIs) for each of the parameters of the fitted model.

(c) Find the probability of failure of the O-rings under the conditions of that day $\mathrm{x}_C = (1, 31, 200)^\top$.

(d) Interpret the results.

Solution. Verify your findings with `https://rpubs.com/FJRubio/Challenger`.

3 Poisson Regression

3.1 The Poisson log-linear model

Example 3.1.1. *Neurons in the central nervous system transmit signals via a series of action potentials, or "spikes". The spiking of a single neuron may be measured by a microelectrode, and its sequence of spikes over time is called a spike train. A simple and commonly-used statistical model for a spike train is an inhomogeneous Poisson point process, which has the following property: For n time windows of length Δ, letting Y_i denote the number of spikes generated by the neuron in the i^{th} time window, the random variables $Y_1, \ldots, Y_n$ are independent and distributed as $Y_i \sim \text{Poisson}(\lambda_i \Delta)$, where the parameter λ_i controls the spiking rate in the i^{th} time window. For simplicity, we will assume $\Delta = 1$.*

The spiking rate λ_i of a neuron may be influenced by external sensory stimuli present in this i^{th} window of time, for example, the intensity and pattern of light visible to the eye or the texture of an object presented to the touch. To understand the effects of these sensory stimuli on the spiking rate of a particular neuron, we may perform an experiment that applies different stimuli in different windows of time and records the neural response. Encoding the stimuli applied in the i^{th} window of time by a set of p covariates $x_{i1}, \ldots, x_{ip}$, a simple model for the Poisson rate parameter λ_i is given by

$$\log \lambda_i = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip} = \mathbf{x_i}'\beta, \tag{4}$$

or equivalently,

$$\mathbb{E}(Y_i|\mathbf{x_i}) = \lambda_i = e^{\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}} = \exp(\mathbf{x_i}'\beta). \tag{5}$$

Together with the distributional assumption $Y_i \sim \text{Poisson}(\lambda_i)$, this is called the **Poisson log-linear model**, or the Poisson regression model. It is a special case of what is known in neuroscience as the linear-nonlinear Poisson cascade model. This area of statistics is popularly known as *count data regression*[2].

More generally, the Poisson log-linear model is a model for n responses $Y_1, \ldots, Y_n$ that take integer count values $(0, 1, 2, \ldots)$. Each Y_i is modeled as an independent $\text{Poisson}(\lambda_i)$ random variable, where $\log \lambda_i$ is a linear combination of the covariates corresponding to the i^{th} observation. As in the cases of linear and logistic regression, we treat the covariates as fixed constants, and the model parameters to be inferred are the regression coefficients $\beta = (\beta_0, \ldots, \beta_p)$.

Note: Loosely speaking, λ_i ranges from 0 to ∞ and $\log(\lambda_i)$ range between $-\infty$ to ∞. That is why the log of the rate parameter is used to avoid modeling a variable with a restricted range.

3.2 Interpretation of coefficients

Differentiating (5) w.r.t. x_{ij}, we get

$$\frac{\partial \mathbb{E}(Y_i|\mathbf{x_i})}{\partial x_{ij}} = \mathbb{E}(Y_i|\mathbf{x_i}) \times \beta_j.$$

Therefore, a one-unit change in the j^{th} regressor leads to a change in the conditional mean by the amount $\mathbb{E}(Y_i|\mathbf{x_i}) \times \beta_j$ (whereas in the linear model we would simply have β_j). In some cases, a regressor may first be transformed by a natural log algorithm.

[2] Ref: Cameron, A. C., & Trivedi, P. K. (2013). Regression analysis of count data. Cambridge University Press.

3.3 Statistical inference

We will describe the procedure for maximum-likelihood estimation of the regression coefficients and Fisher-information-based estimation of their standard errors and discuss some issues concerning model misspecification and robust standard error estimates.

Since $Y_1, \ldots, Y_n$ are independent Poisson random variables, the likelihood function is given by

$$\operatorname{lik}\left(\beta_0, \ldots, \beta_p\right) = \prod_{i=1}^{n} \frac{\lambda_i^{Y_i} e^{-\lambda_i}}{Y_i!},$$

where λ_i is defined in terms of $\beta_0, \ldots, \beta_p$ and the covariates $x_{i1}, \ldots, x_{ip}$ via equation (4). Setting $x_{i0} \equiv 1$ for all i, the log-likelihood is then

$$\begin{aligned} l\left(\beta_0, \ldots, \beta_p\right) &= \sum_{i=1}^{n} Y_i \log \lambda_i - \lambda_i - \log Y_i! \\ &= \sum_{i=1}^{n} Y_i \left(\sum_{j=0}^{p} \beta_j x_{ij} \right) - e^{\sum_{j=0}^{p} \beta_j x_{ij}} - \log Y_i! \end{aligned}$$

and the MLEs are the solutions to the system of score equations, for $m = 0, \ldots, p$,

$$0 = \frac{\partial l}{\partial \beta_m} = \sum_{i=1}^{n} x_{im} \left(Y_i - e^{\sum_{j=0}^{p} \beta_j x_{ij}} \right).$$

These equations may be solved numerically using the Newton-Raphson method.

The Fisher information matrix $I_{\mathbf{Y}}(\beta) = -\mathbb{E}_\beta \left[\nabla^2 l(\beta)\right]$ may be obtained by computing the second-order partial derivatives of l:

$$\frac{\partial^2 l}{\partial \beta_m \partial \beta_l} = -\sum_{i=1}^{n} x_{im} x_{il} e^{\sum_{j=0}^{p} \beta_j x_{ij}}.$$

Writing $X_j = (x_{1j}, \ldots, x_{nj})$ as the j^{th} column of the covariate matrix X and defining the diagonal matrix

$$W = W(\beta) := \operatorname{diag}\left(e^{\sum_{j=0}^{p} \beta_j x_{1j}}, \ldots, e^{\sum_{j=0}^{p} \beta_j x_{nj}} \right),$$

the above may be written as $\frac{\partial^2 l}{\partial \beta_m \partial \beta_l} = -X_m^T W X_l$, so $\nabla^2 l(\beta) = -X^T W X$ and $I_{\mathbf{Y}}(\beta) = X^T W X$. For large n, if the Poisson log-linear model is correct, then the MLE vector $\hat{\beta}$ is approximately distributed as $\mathcal{N}\left(\beta, \left(X^T W X\right)^{-1}\right)$. We may then estimate the standard error of $\hat{\beta}_j$ by

$$\hat{\text{se}}_j = \sqrt{\left(\left(X^T \hat{W} X \right)^{-1} \right)_{jj}},$$

where $\hat{W} = W(\hat{\beta})$ is the plugin estimate for W. These formulas are the same as for the case of logistic regression, except with a different form of the diagonal matrix W.

The modeling assumption of a Poisson distribution for Y_i is rather restrictive, as it implies that the variance of Y_i must be equal to its mean. This is rarely true in practice, and it is frequently the case that the observed variance of Y_i is larger than its mean – this problem is known as **overdispersion**. Nonetheless, the Poisson regression model is oftentimes used in overdispersed settings: As long as $Y_1, \ldots, Y_n$ are independent and

$$\log \mathbb{E}\left[Y_i\right] = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}$$

for each i (so the model for the means of the Y_i''s is correct), then it may be shown that the MLE $\hat{\beta}$ in the Poisson regression model is unbiased for β, even if the distribution of Y_i is not Poisson and the variance of Y_i exceeds its mean. The above standard error estimate $\hat{se}_j$ and the associated confidence interval for β_j, though, would not correct in the overdispersed setting. One may use instead the robust sandwich estimate of the covariance of $\hat{\beta}$, given by

$$\left(X^T \hat{W} X\right)^{-1}\left(X^T \tilde{W} X\right)\left(X^T \hat{W} X\right)^{-1},$$

where

$$\tilde{W} = \operatorname{diag}\left(\left(Y_1 - \hat{\lambda}_1\right)^2, \ldots, \left(Y_n - \hat{\lambda}_n\right)^2\right)$$

and $\hat{\lambda}_i = e^{\sum_{j=0}^{p} \hat{\beta}_j x_{ij}}$ is the fitted value of λ for the i^{th} observation. Alternatively, one may use the pairs bootstrap procedure as described in Logistic Regression.

Remark 3.1. *The linear model, logistic regression model, and Poisson regression model are all examples of the* ***generalized linear model (GLM)****. In a generalized linear model, $Y_1, \ldots, Y_n$ are modeled as independent observations with distributions $Y_i \sim f\left(y \mid \theta_i\right)$ for some one-parameter family $f(y \mid \theta)$. The parameter θ_i is modeled as*

$$g\left(\theta_i\right) = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}$$

for some one-to-one transformation $g: \mathbb{R} \to \mathbb{R}$ called the ***link function****, where $x_{i1}, \ldots, x_{ip}$ are covariates corresponding to Y_i. In the linear model, the parameter was $\theta \equiv \mu$ where $f(y \mid \mu)$ was the* PDF *of the $\mathcal{N}\left(\mu, \sigma_0^2\right)$ distribution (for a known variance σ_0^2), and $g(\mu) = \mu$. In logistic regression, the parameter was $\theta \equiv p$ where $f(y \mid p)$ was the PMF of the Bernoulli (p) distribution, and $g(p) = \log \frac{p}{1-p}$. In Poisson regression, the parameter was $\theta \equiv \lambda$ where $f(y \mid \lambda)$ was the PMF of the Poisson (λ) distribution, and $g(\lambda) = \log \lambda$.*

The choice of the link function g is an important modeling decision, as it determines which transform of the model parameter should be modeled as linear in the observed covariates. In each of the three examples discussed, we used what is called the **natural link**, which is motivated by considering a change-of-variable for the parameter, $\theta \mapsto \eta(\theta)$, so that the PDF/PMF$f(y \mid \eta)$ in terms of the new parameter η has the form

$$f(y \mid \eta) = e^{\eta y - A(\eta)} h(y)$$

for some functions A and h. For example, the Bernoulli PMF is

$$f(y) = p^y (1-p)^{1-y} = (1-p)\left(\frac{p}{1-p}\right)^y = e^{\left(\log \frac{p}{1-p}\right) y + \log(1-p)},$$

so we may set $\eta = \log \frac{p}{1-p}$, $A(\eta) = -\log(1-p) = \log\left(1 + e^\eta\right)$, and $h(y) = 1$. This is called the exponential family form of the PDF/PMF, and η is called the **natural parameter**. In each example, the natural link simply sets $g(\theta) = \eta(\theta)$ (or equivalently, $g(\theta) = c\eta(\theta)$ for a constant c).

Use of the natural link leads to some nice mathematical properties for likelihood-based inference - for instance, since η is modeled as linear in β, the second-order partial derivatives of

$$\log f(Y \mid \eta) = \eta Y - A(\eta) + \log h(Y)$$

with respect to β do not depend on Y, so the Fisher information is always given by $-\nabla^2 l(\beta)$ without needing to take an expectation. (We sometimes say in this case that the "observed and expected Fisher information matrices" are the same.) On the other hand, from the modeling perspective, there is usually no intrinsic reason to believe that the natural link $g(\theta) = \eta(\theta)$ is the correct transformation of θ that is well-modeled as a linear combination of the covariates, and other link functions are also commonly used, especially if they lead to a better fit for the data.

Remark 3.2. *Count data distributions (e.g., visit counts) often have a Poisson distribution, Poisson regression tends to fit these data better than linear regression does (which assumes a normal distribution). There are other alternative count data models in the literature:*

1. **Negative Binomial Model:** *The NB model is very useful if we wish to predict probabilities and not just the mean. It is very useful for overdispersed data but cannot be computed for underdispersed scenarios.*

2. **Zero-inflated Models:** *Zero-inflated models are commonly used to analyze count data, such as the number of visits a patient makes to the emergency room in one year or the number of fish caught in one day in one lake. In some data, the number of zeros is greater than expected using a Poisson or negative binomial distribution. Data with such an excess of zero counts are described as Zero-inflated. ZIP and ZINB emerged as popular choices for those datasets.*

3. **Hurdle Model:** *This model treats the process for zeros differently from that for the counts. Here, the mean of Y_i is no longer $\exp(\mathbf{x_i}'\beta)$, and the Hurdle model can handle both overdispersion and underdispersion.*

Remark 3.3. *Count time series modeling emerged as an area of interest in the late 1970s, with work on the topic rapidly accelerating thereafter. Some key applications and ideas in this field are:*

1. *Count time series arise in numerous applied scientific areas. Examples include the daily number of patients admitted to a hospital, the number of transactions of a given stock observed every minute or the monthly number of car accidents in a region. These data, occasionally observed with some covariates, often share some common characteristics. Foremost, they are frequently dependent as they are observed sequentially in time. As counts, they are integer-valued.*

2. *Count series are often overdispersed (i.e., their variance is greater than their mean), and their autocorrelations are often nonnegative.*

3. *Finally, there are often more zero counts (zero-inflation) than can be explained by the classical marginal count distributions (e.g., Poisson, binomial, negative binomial).*

4. *Good models and analyses should account for these features and provide the user with estimation, fitting, model assessment, prediction, and uncertainty quantification. A detailed review of the available methods is presented here*[3]*.*

[3] Davis, Richard A., et al. "Count time series: A methodological review." JASA (2021).

Hands-on Session. (Fitting a Poisson Regression model)

The GitHub repository (Bicycle Counts Data for NY City) contains a dataset on the number of bicyclists that cross a bridge in New York collected by the New York City Department of Transportation[4]. This data is used to measure bike utilization in transportation planning. Poisson regression is a modeling technique that can be used to predict counts. In this case, we define counts as the number of occurrences of an event in a given period. An equivalent way to look at counts is to look at rates. This session aims to build a Poisson regression model for the number of bicyclists crossing a bridge in New York based on feature variables that include the day of the week, the temperature (high and low), and precipitation. The dataset contains the total daily bike counts conducted monthly on the Brooklyn Bridge, Manhattan Bridge, Williamsburg Bridge, and Queensboro Bridge. For this exercise, we will focus specifically on the Williamsburg Bridge. As a result, we will remove some of the other columns we don't need before working on the modeling part. A brief detail of the column headings of the data is given below:

- Date: Day of the year
- Day: Day of the week
- HighTemp: The daily high temperature (in °F)
- LowTemp: The daily low temperature
- Precipitation: The amount (and type) of precipitation
- Williamsburg Bridge: The Target variable for this regression (count data)

In this study, the data analysis questions we would like you to answer are as follows:

(a) Use explanatory data analysis tools to tackle the non-numeric data in the 'precip' column. Create a box plot to check the distribution of the response variable based on the categorical variable (weekday). Is there any multicollinearity issue in the data?

(b) Fit both the linear regression and Poisson regression models to the data and figure out which one of them suits the situation best.

(c) The above questions lead us to the point where we can predict with our model. Let's say it is a Saturday, with a high temperature of 55 and a low temperature of 28 with no precipitation. What would the Poisson regression model you built predict for a count of bikers on our scaled bridge?

Solution.

(a) First, read the data. Two datasets are available here: Course GitHub. If you work with the raw data ("NYC_Bicycle_Counts_2016.csv"), then you need some cleaning; if the cleaned data ("nyc_bike_counts_cleaned.csv"), then skip the next page.

```
bikes_raw <- read.csv('NYC_Bicycle_Counts_2016.csv',
                      header=T, stringsAsFactors=F)
summary(bikes_raw)
```

We begin by creating a categorical variable for the day of the week. For this exercise, we will focus specifically on the Williamsburg Bridge. As a result, we will remove some of the other columns we don't need as well. The data now has these predictor variables:

[4] NYC Open Data

```
bikes <- data.frame(bikes_raw)
bikes <- data.frame(bikes[c('Day','High.Temp','Low.Temp',
         'Precipitation','WilliamsburgBridge')])
names(bikes) <- c('weekday','hightemp','lowtemp','precip','count')
head(bikes)
```

Here, we can see a potential problem with the data: the precip column has some non-numeric data in it to distinguish (T)race and (S)now types of precipitation. To keep things simple, we will make the following adjustments (We will assume that these bicyclists are tough and NYC bicyclists are still more challenging. Otherwise, why would they be bicycling in NYC!): We will split the precip column into two preciprain and precipsnow. We will put their respective values in each. For trace amounts, we will convert those values to zero.

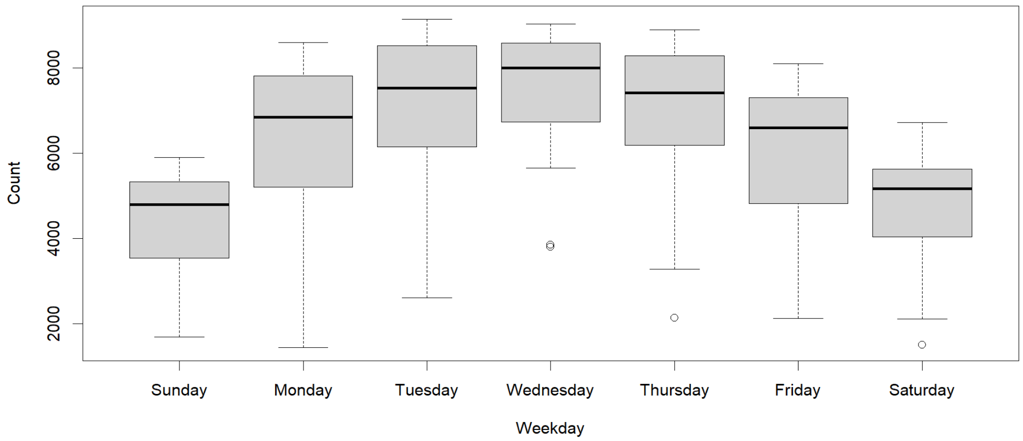
```
na.zero <- function (x) {
    x[is.na(x)] <- 0
    return(x)
}

get.snow <- function(x) {
  result <- 0
  if (grepl('(S)',x)){
    result <- as.double(gsub("[^0-9.-]", "", x))
  }
  return(result)
}

#  Easy part with rain.  This will create a warning, but we will deal with that.
nyc_bikes <- within(bikes, precip_rain<-as.double(precip))
#  Here we deal with the NAs
cat("\n\n")
nyc_bikes$precip_rain <- na.zero(nyc_bikes$precip_rain)

nyc_bikes$precip_snow <- lapply(nyc_bikes$precip, get.snow)
nyc_bikes$precip_snow <- as.numeric(nyc_bikes$precip_snow)
nyc_bikes <- data.frame(nyc_bikes[c('weekday','hightemp','lowtemp',
             'precip_rain','precip_snow','count')])

# Finally, we will convert weekdays to categorical variables.
nyc_bikes$weekday <- as.factor(nyc_bikes$weekday)
head(nyc_bikes)
cat("\n")
summary(nyc_bikes)
write.csv(nyc_bikes, "nyc_bike_counts_cleaned.csv", row.names = F)
```

Finally, we have the cleaned data. Now, we will turn our attention to modeling.

Now, we perform the exploratory analysis by looking at the distribution of the response variable based on the categorical variable, weekday. We will do that with *MASS library in R.*

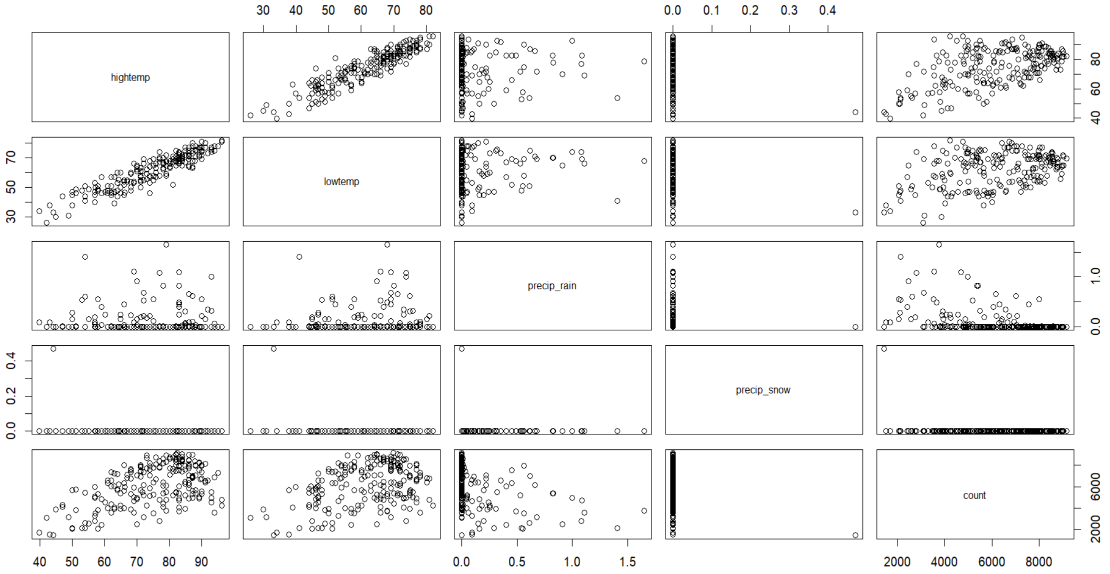
```
library(MASS)
```

```
nyc_bikes$weekday <- factor(nyc_bikes$weekday, levels=c('Sunday', 'Monday',
                    'Tuesday','Wednesday','Thursday','Friday','Saturday'))
boxplot(count~weekday, xlab="Weekday", ylab="Count", data=nyc_bikes)
```



Now, we explore multicollinearity, particularly around temperatures. Also, we note that the snow and rain are mutually exclusive. As part of this analysis, we will see which variables add value and may choose to remove those later. Also, we notice a couple of outliers for snow. We will come back to those concerns later.

```
nyc_bikes_quant=data.frame(nyc_bikes[c('hightemp','lowtemp',
                             'precip_rain','precip_snow','count')])
plot(nyc_bikes_quant)
```

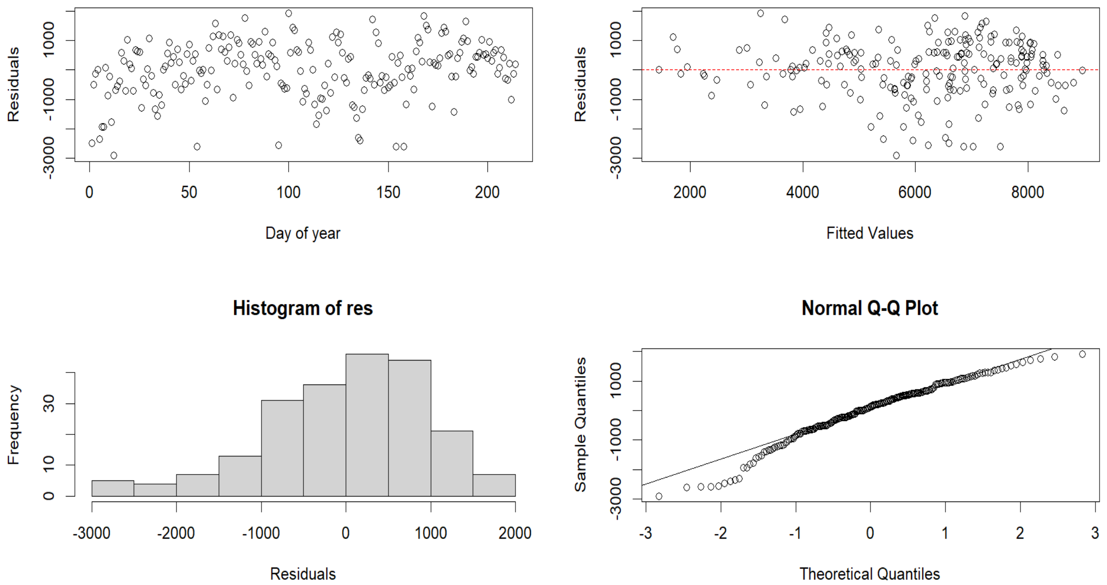


(b) Next, we will fit a linear model to see what we get. This will likely not be a great fit, but we can compare it with a Poisson model.

```
linear_model <- lm(count~., data=nyc_bikes)
plt_model <- function(model)
  {
  par(mfrow=c(2,2))
  res <- model$residuals
  plot(res,mdl$data[1], xlab="Day of year", ylab="Residuals")
  plot(fitted(model), res, xlab="Fitted Values", ylab="Residuals")
  abline(h=0, lty=2, col="red")
  hist(res, xlab="Residuals", ylab="Frequency")
  qqnorm(res)
  qqline(res)
  }
summary(linear_model)
```

```
## Call:
## lm(formula = count ~ ., data = nyc_bikes)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2905.1  -521.7   137.0   619.2  1918.9
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -867.51     444.39  -1.952  0.05230 .
## weekdayMonday      2045.56     252.87   8.089 5.38e-14 ***
## weekdayTuesday     2632.15     252.61  10.420  < 2e-16 ***
## weekdayWednesday   2861.68     252.69  11.325  < 2e-16 ***
## weekdayThursday    2415.68     251.93   9.589  < 2e-16 ***
## weekdayFriday      1756.22     250.51   7.011 3.44e-11 ***
## weekdaySaturday     401.70     249.46   1.610  0.10889
## hightemp            125.14      14.12   8.863 3.98e-16 ***
## lowtemp             -61.30      15.20  -4.034 7.75e-05 ***
## precip_rain       -2358.20     270.75  -8.710 1.07e-15 ***
## precip_snow       -6866.81    2157.81  -3.182  0.00169 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 981.5 on 203 degrees of freedom
## Multiple R-squared:  0.7485, Adjusted R-squared:  0.7361
## F-statistic: 60.41 on 10 and 203 DF,  p-value: < 2.2e-16
```

```
plt_model(linear_model)
```



Let us now compare the above with a Poisson regression on the data. We use a built-in function; however, we already know how to code it from scratch from our earlier experience with logistic regression models.

```
poisson_model <- glm(count~., family="poisson", data=nyc_bikes)
summary(poisson_model)
```

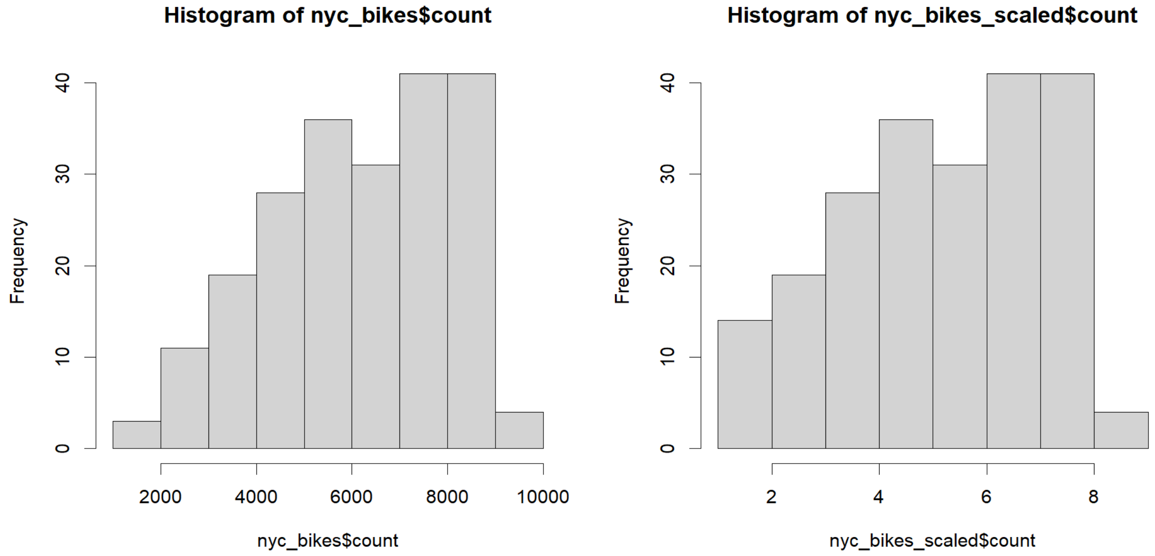
```
## Call:
## glm(formula = count ~ ., family = "poisson", data = nyc_bikes)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -39.545   -7.662    0.784    8.717   27.325
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          7.4386802  0.0064331 1156.32   <2e-16 ***
## weekdayMonday        0.3927792  0.0035492  110.67   <2e-16 ***
## weekdayTuesday       0.4802382  0.0035055  137.00   <2e-16 ***
## weekdayWednesday     0.4990051  0.0034490  144.68   <2e-16 ***
## weekdayThursday      0.4374520  0.0034980  125.06   <2e-16 ***
## weekdayFriday        0.3420589  0.0035907   95.26   <2e-16 ***
## weekdaySaturday      0.0952104  0.0037565   25.35   <2e-16 ***
## hightemp             0.0224908  0.0001900  118.40   <2e-16 ***
## lowtemp             -0.0113279  0.0002006  -56.47   <2e-16 ***
## precip_rain         -0.4931333  0.0043281 -113.94   <2e-16 ***
## precip_snow         -2.5020266  0.0565319  -44.26   <2e-16 ***
## ---
## (Dispersion parameter for poisson family taken to be 1)
```

(c) The Poisson regression model fits better than the linear regression model based on the statistical significance of the coefficients. However, interpreting the coefficients becomes extremely difficult since we worked with actual count data without transforming it. For example, the coefficient for hightemp is 0.0224908. For the hightemp value, the expected count of the number of bike rides per day for one unit increase in hightemp would be:

$$\exp(0.0224908) = 1.0227457.$$

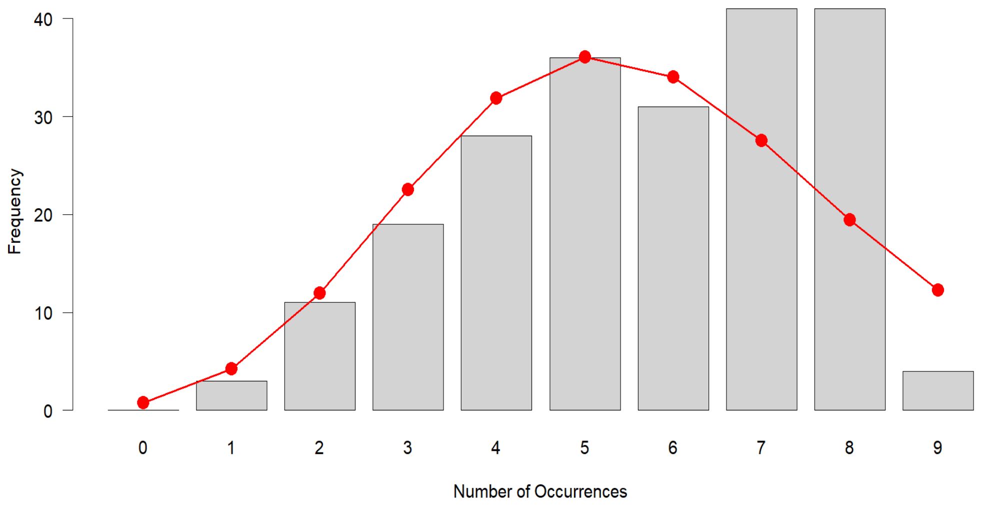
These numbers are a little weird. To avoid this, we transform our response variable. Here, log transformation is not very useful; we divide the response variable by 1000 to scale the data.

```
nyc_bikes_scaled <- data.frame(nyc_bikes)
nyc_bikes_scaled$count<- as.integer(nyc_bikes_scaled$count/1000)
head(nyc_bikes_scaled)

# Histogram of actual count data and scaled data
par(mfrow=c(1,2))
hist(nyc_bikes$count)
hist(nyc_bikes_scaled$count)
```



The distribution of our response data did not change much. We can also see how well a Poisson distribution fits our response variable using *vcd R library*.

```
install.packages("vcd")
library(vcd)
gf <- goodfit(nyc_bikes_scaled$count, "poisson")
plot(gf, type="standing", scale="raw")
```



```
updated_poisson_model <- glm(count~., family="poisson", data=nyc_bikes_scaled)
summary(updated_poisson_model)
```

We obtained our desired model. For Poisson regression, as with logistic regression, we don't have residuals to analyze. However, one can approximate the residuals using the deviance residuals. We can look at the linearity assumption as follows[5]:

```
rsid <- residuals(poisson_model)
d <- deviance(poisson_model)
par(mfrow=c(2,2))
plot(nyc_bikes_scaled$hightemp, rsid)
boxplot(rsid~weekday,xlab="Weekday",ylab = "Std residuals",
       data = nyc_bikes_scaled)
qqnorm(rsid, ylab="Std residuals")
qqline(rsid,col="blue",lwd=2)
hist(rsid)
```

(d)
```
new_data <- data.frame(weekday='Saturday', hightemp=55, lowtemp=28,
                 precip_rain=0, precip_snow=0)
new_data$weekday <- as.factor(new_data$weekday)

pred <- predict(updated_poisson_model, newdata = new_data,
        interval=c('prediction'), lvl=0.9, type="response")
round(pred*1000)
```

The required answer to the question is 4232.

[5] Ref: https://rpubs.com/fractalbass/607394

4 The Proportional Hazards Model

Example 4.0.1. *A clinical trial is performed to study the effect of a drug for maintaining/prolonging remission induced by chemotherapy in the treatment of acute leukemia. (Remission is the disappearance of leukemic cells and other symptoms of the disease.) For each i^{th} patient in the trial, let T_i denote the length of the remission (or equivalently, the time until recurrence of cancer), which we wish to model in terms of patient-specific covariates $x_{i1}, \ldots, x_{ip}$. The first covariate x_{i1} may be a $0-1$ variable indicating whether patient i received the drug or a placebo, and the remaining covariates are other factors, such as the age of the patient, that may affect the remission length.*

Modeling T_i as a continuous, positive-valued random variable with CDF $F_i(t)$ *and* PDF $f_i(t) = F_i'(t)$, *it is useful to think about the distribution of T_i in terms of its **hazard function** $\lambda_i(t)$, which represents the "instantaneous risk" of recurrence at time t:*

$$\lambda_i(t) := \lim_{\delta \to 0} \frac{1}{\delta} \mathbb{P}\left[T_i \leq t + \delta \mid T_i \geq t\right].$$

In other words, for small δ, the probability that a recurrence of cancer occurs in the time window $[t, t+\delta]$, conditional on it not having occurred up to time t, is approximately $\delta\lambda_i(t)$. The hazard function may be expressed in terms of the CDF $F_i(t)$ and PDF $f_i(t)$ *as*

$$\lambda_i(t) = \lim_{\delta \to 0} \frac{\mathbb{P}\left[t \leq T_i \leq t + \delta\right]}{\delta \mathbb{P}\left[T_i \geq t\right]} = \lim_{\delta \to 0} \frac{F_i(t+\delta) - F_i(t)}{\delta\left(1 - F_i(t)\right)} = \frac{f_i(t)}{1 - F_i(t)}.$$

To develop some intuition for the hazard function, consider a simple example where $T_i \sim$ Exponential(θ). *Then the* PDF *is $f_i(t) = \theta e^{-\theta t}$, the* CDF *is $F_i(t) = 1 - e^{-\theta t}$, so the hazard function is*

$$\lambda_i(t) = \frac{\theta e^{-\theta t}}{1 - (1 - e^{-\theta t})} = \theta.$$

In this case, the hazard function is constant in time (which is a special property of the exponential distribution). Intuitively, this means that assuming the remission has lasted until time t, the probability of the recurrence occurring in the next instant of time is the same for every t and is determined only by θ. The parameter θ governs how quickly the exponential distribution decays - the larger the value of θ, the faster the rate of decay, and the more likely it is that the recurrence of cancer will occur at any next instant of time.

Cox's **proportional hazards model** does not assume that the distribution of T_i is exponential, or that it follows any particular parametric form. Instead, it models the hazard function for T_i as

$$\lambda_i(t) = \lambda_0(t) \exp\left(\beta_1 x_{i1} + \ldots + \beta_p x_{ip}\right)$$

The regression coefficients $\beta_1, \ldots, \beta_p$ are unknown parameters determining the effects of the covariates on the remission length T_i, and $\lambda_0(t)$ is a completely unknown **baseline hazard function**. In this model, $\lambda_0(t)$ controls the shape of the hazard function over time for all patients, and the factor $\exp\left(\beta_1 x_{i1} + \ldots + \beta_p x_{ip}\right)$ controls the scale of the hazard function for each patient i. Thus the model asserts that for any two patients i and j, their hazard functions have the same shape and differ only in scale so that the ratio of their hazard functions $\lambda_i(t)/\lambda_j(t)$ is constant over time (hence the name "proportional" hazards). The model posits that this scale ratio is determined by a linear combination of the differences of the patients' covariates.

In the clinical trial, the remission for a patient i may last longer than the duration for which the patient participates in the trial. In this case, we do not observe their true remission length T_i (which can take the value ∞ if the cancer never returns), but instead, we only observe that $T_i > l_i$ where l_i is the length of time for which the patient is in the trial. This type of observation is called **right-censored**. The method of inference developed below for the proportional hazards model will naturally handle data in which some of the observations are right-censored. We treat l_i as a fixed and known constant for every patient, so that we either observe a value of T_i that is at most l_i, or we observe that $T_i > l_i$. We will make an important assumption that T_i (the true remission length) does not depend on l_i (the right-censoring time).

The proportional hazards model may be used to model the time-to-onset of any event pertaining to an individual in terms of observed covariates for that individual; example applications include medical trials as above, as well as industrial reliability experiments that model the time-to-failures of devices. According to a 2014 list in the scientific journal *Nature*, the 1972 paper by David Cox that introduced this model is the 2^{nd} most cited paper in statistics and the 24^{th} most cited paper in all of science.

4.1 Statistical inference

In many applications, the regression coefficients $\beta_1, \ldots, \beta_p$ are of greater interest than the baseline hazard function $\lambda_0(t)$. If the first covariate x_{i1} corresponds to an indicator variable representing assignment to the treatment group (drug) or the control group (placebo), then the coefficient β_1 represents the log-hazards-ratio between the two groups after controlling for the other covariates $x_{i2}, \ldots, x_{ip}$. We will discuss inference procedures for the following tasks:

- Estimate $\beta_1, \ldots, \beta_p$.
- Test whether $\beta_1 = 0$.

Perhaps surprisingly, it is possible to perform these inference tasks without any knowledge of, and without any assumptions regarding, the baseline hazard function $\lambda_0(t)$.

In previous models, we performed inference by writing down the likelihood of the model parameters. Inference in the proportional hazards model will be slightly different from these previous examples, because the baseline hazard function $\lambda_0(t)$ is completely unknown, and the likelihood function and MLEs for $\beta_1, \ldots, \beta_p$ would depend on $\lambda_0(t)$. If $\lambda_0(t)$ were modeled parametrically using a small number of additional parameters, then we may include these as parameters of the model and fit the entire model by computing the joint MLEs of these additional parameters and $\beta_1, \ldots, \beta_p$. However, without parametric modeling assumptions on $\lambda_0(t)$, in this course, we have not discussed procedures for how to estimate an entire unknown function $\lambda_0(t)$.

We will circumvent this problem by conditioning on the set of all distinct observed recurrence times $t_{(1)} < t_{(2)} < \ldots < t_{(m)}$ across all patients. (This idea is quite similar to how we conditioned on the set of all distinct observed values in permutation two-sample tests, to address the problem of an unknown common distribution function $F = G$ under the null hypothesis.) Since we are modeling T_i as continuous random variables, we may assume that each observed recurrence time $t_{(k)}$ corresponds to only one patient (i.e. there are no ties in recurrence times), so m is just the total number of non-right-censored observations. For each $t_{(k)}$, the *risk set* $\mathcal{R}_{(k)}$ immediately before time $t_{(k)}$ is the set of patients who have not yet left the study (been right-censored) and are still in remission-this represents the candidate set of patients for which we may have observed a recurrence at time $t_{(k)}$. Conditional on the fact that some patient in this risk set $\mathcal{R}_{(k)}$ has a recurrence at time $t_{(k)}$, the probability that it is a particular patient $I_k \in \mathcal{R}_{(k)}$ is

$$\frac{\lambda_{I_k}\left(t_{(k)}\right)}{\sum_{i \in \mathcal{R}_{(k)}} \lambda_i\left(t_{(k)}\right)}$$

(the ratio of the "instantaneous rate" of recurrence for patient I_k to the sum of the rates for all candidate patients). Under the proportional hazards model, this is

$$\frac{\lambda_0\left(t_{(k)}\right) \exp\left(\beta_1 x_{I_k 1} + \ldots + \beta_p x_{I_k p}\right)}{\sum_{i \in \mathcal{R}_{(k)}} \lambda_0\left(t_{(k)}\right) \exp\left(\beta_1 x_{i1} + \ldots + \beta_p x_{ip}\right)}.$$

Importantly, the factor $\lambda_0\left(t_{(k)}\right)$ cancels from the numerator and denominator of this expression, yielding a quantity that does not depend on the baseline hazard function $\lambda_0(t)$. Taking a product of the above expression overall observed recurrence times yield

$$\operatorname{plik}\left(\beta_1, \ldots, \beta_p\right) = \prod_{k=1}^{m} \frac{\exp\left(\beta_1 x_{I_k 1} + \ldots + \beta_p x_{I_k p}\right)}{\sum_{i \in \mathcal{R}_{(k)}} \exp\left(\beta_1 x_{i1} + \ldots + \beta_p x_{ip}\right)}$$

This quantity is called the **partial likelihood function** of $\beta_1, \ldots, \beta_p$. Intuitively, it captures all of the information contained by the observations that at each time $t_{(k)}$, the particular recurrence was for patient I_k as opposed to the other patients for which we could have observed a recurrence at that time. We may perform likelihood-based inference using this partial likelihood in place of the usual likelihood function. We may estimate $\beta_1, \ldots, \beta_p$ by maximizing the partial likelihood over these parameters. As with usual MLE calculations, it is computationally convenient first to take a logarithm, so we consider the log-partial likelihood

$$l\left(\beta_1, \ldots, \beta_p\right) = \sum_{k=1}^{m} \left(\beta_1 x_{I_k 1} + \ldots + \beta_p x_{I_k p} - \log \sum_{i \in \mathcal{R}_{(k)}} \exp\left(\beta_1 x_{i1} + \ldots + \beta_p x_{ip}\right) \right).$$

We may maximize this quantity by setting its derivative with respect to each $\beta_1, \ldots, \beta_p$ equal to 0:

$$0 = \frac{\partial l}{\partial \beta_j} = \sum_{k=1}^{m} \left(x_{I_k j} - \frac{\sum_{i \in \mathcal{R}_{(k)}} x_{ij} \exp\left(\beta_1 x_{i1} + \ldots + \beta_p x_{ip}\right)}{\sum_{i \in \mathcal{R}_{(k)}} \exp\left(\beta_1 x_{i1} + \ldots + \beta_p x_{ip}\right)} \right).$$

Solving numerically this system of p equations in p unknowns $\beta_1, \ldots, \beta_p$ yields the maximum partial-likelihood estimates $\hat{\beta}_1, \ldots, \hat{\beta}_p$.

The asymptotic theory for the maximum partial-likelihood estimate is analogous to that of the MLE in usual parametric models (although the mathematical derivation of this theory requires more advanced probabilistic tools that we did not cover in this course). In particular, the usual generalized likelihood ratio test applies: To test $H_0 : \beta_1 = 0$, we may compute the maximum partial likelihood estimates $\hat{\beta}_{2,0}, \ldots, \hat{\beta}_{p,0}$ in this sub-model, using the same procedure as above with the first covariate removed. The test statistic

$$-2 \log \Lambda = -2 \log \frac{\operatorname{plik}\left(0, \hat{\beta}_{2,0}, \ldots, \hat{\beta}_{p,0}\right)}{\operatorname{plik}\left(\hat{\beta}_1, \ldots, \hat{\beta}_p\right)}$$

is, under mild regularity conditions, distributed as χ_1^2 in the limit of large n, and an asymptotic level-α test would reject H_0 when $-2 \log \Lambda$ exceeds the upper-α point $\chi_1^2(\alpha)$.

5 The Bradley-Terry model

Example 5.0.1. *There are 30 basketball teams in the NBA, each playing 82 games in the regular season (so there are 1230 total games). At the end of the regular season, we observe which two teams (i, j) played in each game and whether team i or team j won. How can we rank the teams and/or determine the strength of each team?*[6]

The simplest strategy might be to compare the number of games won by each team. However, the NBA season is structured so that every team plays every other team a different number of times (between 2 and 4). So, the teams have different "strengths of schedule", meaning that some teams play stronger opponents more frequently than other teams. These teams might have worse win-loss records but, in fact, be better than other teams that won more games against weaker opponents.

*A model-based approach to address this problem is the following: Let $\beta_i \in \mathbb{R}$ represent the "strength" of team i, and let the outcome of a game between teams (i, j) be determined by $\beta_i - \beta_j$. The **Bradley-Terry model** treats this outcome as an independent Bernoulli random variable with distribution* $\text{Bernoulli}(p_{ij})$*, where the log-odds corresponding to the probability p_{ij} that team i beats team j is modeled as*

$$\log \frac{p_{ij}}{1 - p_{ij}} = \beta_i - \beta_j.$$

Equivalently, solving for p_{ij} yields

$$p_{ij} = \frac{e^{\beta_i - \beta_j}}{1 + e^{\beta_i - \beta_j}} = \frac{e^{\beta_i}}{e^{\beta_i} + e^{\beta_j}}.$$

This model is over-parametrized in the sense that it is exactly the same if we add a fixed constant c to all values β_i because the differences $\beta_i - \beta_j$ remain unchanged. We may fix this problem by setting $\beta_i \equiv 0$ for a particular team, for example, $\beta_{Warriors} \equiv 0$. Then, for every other team $j, \beta_j = \beta_j - 0$ represents the log-odds that team j beats the Warriors.

If we always order each pair (i, j) so that team i is the home team and j is the away team, then we may incorporate a home-court advantage by including an intercept term α:

$$\log \frac{p_{ij}}{1 - p_{ij}} = \alpha + \beta_i - \beta_j,$$

or equivalently

$$p_{ij} = \frac{e^{\alpha + \beta_i - \beta_j}}{1 + e^{\alpha + \beta_i - \beta_j}}. \tag{6}$$

This increases the log-odds of the home team winning in every game by a constant value α.

More generally, the Bradley-Terry model assigns scores to a fixed set of items based on pairwise comparisons of these items, where the log-odds of item i "beating" item j is given by the difference of their scores. An intercept term may be included to account for a systematic difference between the first and second items of each comparison. Applications of the model include the ranking of competitors in sports and chess, the ranking of products in paired comparison surveys of consumer choice, analysis of dominance hierarchies within animal and human communities, and estimation of the relevance of documents in machine-learned search engines (JMLR, 2023).

[6] Reference: STAT200 Course (`https://web.stanford.edu/class/stats200/lectures.html`)

5.1 Statistical inference

Let $k = 30$ be the number of NBA teams, and denote the Warriors as team 1. We might be interested in the following inferential tasks:

- Estimate the home-court advantage α and the team strengths $\beta_1, \ldots, \beta_k$ (constraining, say, $\beta_1 = \beta_{\text{Warriors}} \equiv 0$).
- Test the null hypothesis of no home-court effect, $\alpha = 0$.
- Obtain a confidence interval for $\beta_i - \beta_j$ for two particular teams (i, j).

Suppose we observe n total games $(i_1, j_1), \ldots, (i_n, j_n)$ between these k teams, where each (i, j) is a pair of distinct teams in $\{1, \ldots, k\}$ and the home team is team i. Let $Y_1, \ldots, Y_n \in \{0, 1\}$ be such that $Y_m = 1$ if i_m beat j_m in the m^{th} game and $Y_m = 0$ otherwise. The likelihood for the parameters $\theta = (\alpha, \beta_2, \ldots, \beta_k)$ is then given by

$$\operatorname{lik}(\alpha, \beta_2, \ldots, \beta_k) = \prod_{m=1}^{n} p_{i_m j_m}^{Y_m} \left(1 - p_{i_m j_m}\right)^{1-Y_m} = \prod_{m=1}^{n} \left(1 - p_{i_m j_m}\right) \left(\frac{p_{i_m j_m}}{1 - p_{i_m j_m}}\right)^{Y_m},$$

where p_{ij} is given as a function of α, β_i, and β_j by equation (6) and we set $\beta_1 \equiv 0$. The log-likelihood is

$$\begin{aligned} l(\alpha, \beta_2, \ldots, \beta_k) &= \sum_{m=1}^{n} Y_m \log\left(\frac{p_{i_m j_m}}{1 - p_{i_m j_m}}\right) + \log\left(1 - p_{i_m j_m}\right) \\ &= \sum_{m=1}^{n} Y_m \left(\alpha + \beta_{i_m} - \beta_{j_m}\right) - \log\left(1 + e^{\alpha + \beta_{i_m} - \beta_{j_m}}\right). \end{aligned} \tag{7}$$

To estimate the parameters $\theta = (\alpha, \beta_2, \ldots, \beta_k)$ using the MLE, we set the partial derivative with respect to each parameter $\alpha, \beta_2, \ldots, \beta_k$ equal to 0 :

$$0 = \frac{\partial l}{\partial \alpha} = \sum_{m=1}^{n} Y_m - \frac{e^{\alpha + \beta_{i_m} - \beta_{j_m}}}{1 + e^{\alpha + \beta_{i_m} - \beta_{j_m}}} \tag{8}$$

$$0 = \frac{\partial l}{\partial \beta_i} = \sum_{m: i_m = i} \left(Y_m - \frac{e^{\alpha + \beta_{i_m} - \beta_{j_m}}}{1 + e^{\alpha + \beta_{i_m} - \beta_{j_m}}}\right) + \sum_{m: j_m = i} \left(-Y_m + \frac{e^{\alpha + \beta_{i_m} - \beta_{j_m}}}{1 + e^{\alpha + \beta_{i_m} - \beta_{j_m}}}\right). \tag{9}$$

This yields a system of k equations in the k unknowns $\alpha, \beta_2, \ldots, \beta_k$, which may be solved numerically using the Newton-Raphson algorithm. The solution is the MLE $\hat{\theta} = \left(\hat{\alpha}, \hat{\beta}_2, \ldots, \hat{\beta}_k\right)$.

To test the null hypothesis $H_0 : \alpha = 0$, we may use the generalized likelihood ratio test (GLRT): Under the sub-model where $\alpha = 0$, the log-likelihood function is

$$l(\beta_2, \ldots, \beta_k) = \sum_{m=1}^{n} Y_m \left(\beta_{i_m} - \beta_{j_m}\right) - \log\left(1 + e^{\beta_{i_m} - \beta_{j_m}}\right),$$

and the system of score equations satisfied by the sub-model MLE is

$$0 = \frac{\partial l}{\partial \beta_i} = \sum_{m: i_m = i} \left(Y_m - \frac{e^{\beta_{i_m} - \beta_{j_m}}}{1 + e^{\beta_{i_m} - \beta_{j_m}}}\right) + \sum_{m: j_m = i} \left(-Y_m + \frac{e^{\beta_{i_m} - \beta_{j_m}}}{1 + e^{\beta_{i_m} - \beta_{j_m}}}\right)$$

for $i = 2, \dots, k$. We may solve these equations using Newton-Raphson to obtain the submodel MLEs $\hat{\beta}_{2,0}, \dots, \hat{\beta}_{k,0}$. The GLRT of $\alpha = 0$ is based on the test statistic.

$$-2 \log \Lambda = -2 \log \frac{\text{lik}\left(0, \hat{\beta}_{2,0}, \dots, \hat{\beta}_{k,0}\right)}{\text{lik}\left(\hat{\alpha}, \hat{\beta}_2, \dots, \hat{\beta}_k\right)},$$

and an approximate level-0.05 test rejects H_0 when $-2 \log \Lambda > \chi^2_1(0.05)$. (The number of degrees of freedom is 1 because the full model has one more parameter, α, than the submodel.)

We may obtain a confidence interval for $\beta_i - \beta_j$ by centering it around $\hat{\beta}_i - \hat{\beta}_j$, and estimating the standard error of $\hat{\beta}_i - \hat{\beta}_j$. Let us first consider the sampling distribution of the entire vector of MLE estimates $\hat{\theta} = \left(\hat{\alpha}, \hat{\beta}_2, \dots, \hat{\beta}_k\right)$. When the number of total games n is large, this is approximately $\mathcal{N}\left(\theta, I_{\mathbf{Y}}(\theta)^{-1}\right)$, where $I_{\mathbf{Y}}(\theta) = -\mathbb{E}_\theta\left[\nabla^2 l(\theta)\right]$. The Hessian matrix $\nabla^2 l(\theta)$ may be computed by differentiating the right sides of the score equations (8) and (9) a second time with respect to the variables $\alpha, \beta_2, \dots, \beta_k$. (We will do this explicitly for the more general logistic regression model.) It is easy to see that $\nabla^2 l(\theta)$ is a constant quantity that does not involve $Y_1, \dots, Y_n$, so $I_{\mathbf{Y}}(\theta) = -\nabla^2 l(\theta)$.

Finally, since $\hat{\beta}_i - \hat{\beta}_j$ is a linear combination of the coordinates of $\hat{\theta}$, it is approximately normal when $\hat{\theta}$ is approximately multivariate normal. Its mean is $\mathbb{E}\left[\hat{\beta}_i - \hat{\beta}_j\right] \approx \beta_i - \beta_j$, and its variance is

$$\begin{aligned}
\text{Var}\left[\hat{\beta}_i - \hat{\beta}_j\right] &= \text{Cov}\left[\hat{\beta}_i - \hat{\beta}_j, \hat{\beta}_i - \hat{\beta}_j\right] \\
&= \text{Var}\left[\hat{\beta}_i\right] + \text{Var}\left[\hat{\beta}_j\right] - 2\,\text{Cov}\left[\hat{\beta}_i, \hat{\beta}_j\right] \\
&\approx \left(I_{\mathbf{Y}}^{-1}(\theta)\right)_{ii} + \left(I_{\mathbf{Y}}^{-1}(\theta)\right)_{jj} - 2\left(I_{\mathbf{Y}}^{-1}(\theta)\right)_{ij}.
\end{aligned}$$

We may estimate the standard error of $\hat{\beta}_i - \hat{\beta}_j$ by the plug-in estimate

$$\hat{\mathbf{se}}_{ij} = \sqrt{\left(I_{\mathbf{Y}}^{-1}(\hat{\theta})\right)_{ii} + \left(I_{\mathbf{Y}}^{-1}(\hat{\theta})\right)_{jj} - 2\left(I_{\mathbf{Y}}^{-1}(\hat{\theta})\right)_{ij}}.$$

A 95% confidence interval for $\beta_i - \beta_j$, assuming correctness of the Bradley-Terry model, is then given by $\hat{\beta}_i - \hat{\beta}_j \pm z(0.025)\hat{\text{se}}_{ij}$.

Hands-on Session. (Fitting a Bradley-Terry model)

The GitHub repository NBA Data contains the results of all 1230 NBA games from the 2015-2016 regular season. The 30 teams are encoded numerically from 1 to 30; the key for this encoding is provided in the file *NBA_teams.txt*. Each row of *NBA_record.csv* indicates the home team, away team, and outcome Y for one game, where $Y = 1$ if the home team won and $Y = 0$ otherwise. For parts (a) and (b), you may *not* use an existing software implementation of the Bradley-Terry or logistic regression model; however, you may use any generic optimization or equation-solving routine (or you may implement the Newton-Raphson iterations yourself)[7].

(a) Fit the Bradley-Terry model, with an intercept term α for the home-court advantage, to this data set. What are the 8 teams (in ranked order) with the highest Bradley-Terry scores? How much greater are the log odds of winning for the home team than for the away team?

[7] `https://www.r-bloggers.com/2022/02/what-is-the-bradley-terry-model/`

One approach to do this in R is to use the generic optimization function optim. To do this, first define a function

```
loglik = function(theta,Home,Away,Y) {
         ...
   }
```

that returns the log-likelihood for the Bradley-Terry model given inputs $\theta = (\alpha, \beta_2, \ldots, \beta_k)$ (constraining $\beta_1 \equiv 0$), Home $= (i_1, \ldots, i_n)$, Away $= (j_1, \ldots, j_n)$, and Y $= (Y_1, \ldots, Y_n)$, where i_m and j_m are the home and away teams for game m. To then read the data file and maximize the log-likelihood:

```
table = read.csv("NBA_record.csv")
result = optim(theta0,loglik,Home=table$Home,Away=table$Away,Y=table$Y,
               method="BFGS",control=list("fnscale"=-1))
```

Here theta0 is any initialization for θ (for example the all 0's vector). The method will use the BFGS algorithm, and "fnscale" $= -1$ indicates that it should perform maximization rather than minimization.

(b) Fit the Bradley-Terry model without an intercept term. (You may do this in R by defining a new function

```
loglik_noalpha = function(theta,Home,Away,Y)
```

where now $\theta = (\beta_2, \ldots, \beta_k)$, and using optim as before.) Evaluate the log-likelihoods at the full model and sub-model MLEs, and conduct a generalized likelihood ratio test of the null hypothesis of no home court advantage, $H_0 : \alpha = 0$. What is the p-value that you obtain for your test?

(c) For the m^{th} game, suppose we define 30 covariates $x_{m,1}, \ldots, x_{m,30}$ in the following way: Let $x_{m,1} = 1$ always. Let $x_{m,i} = 1$ if team i is the home team of this game and $i \neq 1$, and let $x_{m,j} = -1$ if team j is the away team of this game and $j \neq 1$. Let $x_{m,k} = 0$ for all other k. Explain why logistic regression for Y_m using the covariates $x_{m,1}, \ldots, x_{m,30}$ is equivalent to the Bradley-Terry model, where we constrain the Bradley-Terry score of team 1 to be $\beta_1 \equiv 0$. If we were to run this logistic regression, what would be the interpretation of the fitted coefficient for the first covariate $x_{m,1}$? For the 10th covariate $x_{m,10}$?

(d) Fit the logistic regression in part (c) using any standard regression software, and verify that the fitted coefficients match (up to reasonable numerical accuracy) your estimated parameters from part (a).

To do this in R, you may construct a matrix X of size 1230×30 containing the covariates as defined in part (c) and then fit the regression using

```
model = glm.fit(X,table$Y,family=binomial())
coefs = model$coefficients
```

Solution.

(a) First, read the data.

```
table = read.csv('NBA_record.csv')
teams = read.csv('NBA_teams.txt', header=FALSE, as.is=TRUE)
num_games = nrow(table)
num_teams = nrow(teams)
```

The Bradley-Terry log-likelihood, as defined in Class Note, is

$$\sum_{m=1}^{n} Y_m \left(\alpha + \beta_{i_m} - \beta_{j_m}\right) - \log\left(1 + e^{\alpha+\beta_{i_m}-\beta_{j_m}}\right)$$

Let's write a function to compute this.

```
loglik = function(theta, Home, Away, Y) {
  alpha = theta[1]
  beta = c(0, theta[-1])
  params = alpha + beta[Home]- beta[Away]
  return(sum(Y * params- log(1 + exp(params))))
}
```

We can train the model using the *optim* function.

```
theta0 = rep(0, num_teams)
result = optim(theta0, loglik,
               Home=table$Home, Away=table$Away, Y=table$Y,
               method='BFGS', control=list('fnscale'=-1))
```

Now we print the 8 teams with the highest Bradley-Terry scores:

```
coefs = c(0, result$par[-1])
ranking = order(coefs, decreasing=TRUE)
data.frame(team=teams[ranking[1:8],],
           score=coefs[ranking[1:8]])
```

```
##                         team       score
## 1 10 Golden State Warriors  1.90274553
## 2      27 San Antonio Spurs  1.29320880
## 3    6 Cleveland Cavaliers  0.48246253
## 4        28 Toronto Raptors  0.42940629
## 5 21 Oklahoma City Thunder  0.42045913
## 6   13 Los Angeles Clippers  0.28898908
## 7          1 Atlanta Hawks  0.00000000
## 8            16 Miami Heat -0.01215086
```

The intercept term indicates the home-court advantage:

```
result$par[1]
```

```
## [1] 0.4626864
```

So the home team has a $\hat{\alpha} = 0.4626864$ greater log-odds of winning than the away team. This means, for example, that we can predict the home team has a $e^{-\hat{\alpha}} = 0.62959$ chance of winning if the two teams are evenly matched.

(b) Let's define the log-likelihood function without an intercept and run the optimization.

```
loglik_noalpha = function(theta, Home, Away, Y) {
  beta = c(0, theta)
  params = beta[Home]- beta[Away]
  return(sum(Y * params- log(1 + exp(params))))
  }

theta0 = rep(0, num_teams- 1)
result_noalpha = optim(theta0, loglik_noalpha,
                       Home=table$Home, Away=table$Away, Y=table$Y,
                       method='BFGS', control=list('fnscale'=-1))
```

We can take a look at the optimal log-likelihood values:

```
print(result$value)

## [1]-680.2417

print(result_noalpha$value)

## [1]-705.08
```

Now we can perform a generalized likelihood ratio test, using a χ^2 cutoff with 1 degree of freedom.

```
statistic =-2 * (result_noalpha$value - result$value)
p_value = 1- pchisq(statistic, df=1)
print(statistic)

## [1] 49.67658

print(p_value)

## [1] 1.812994e-12
```

The p-value is extremely small, so we can safely reject the null hypothesis that there is no home court advantage.

(c) Logistic regression models the log odds of the probability of winning as linear in the covariates,

$$\log \frac{p_m}{1-p_m} = \alpha + \beta_1 x_{m,1} + \cdots + \beta_{30}, x_{m,30},$$

Taking the specified design matrix reduces the above expression to

$$\log \frac{p_m}{1-p_m} = \alpha + \beta_{i_m} - \beta_{j_m}$$

where i_m and j_m are the indices for the home and away teams that played in game m, respectively. From here, we can see that $p_m = p_{i_m j_m}$, the Bradley-Terry specification of the probability of team i_m beating team j_m. Above, the interpretation of the first estimated coefficient is the home court advantage and the interpretation of the tenth coefficient is the log-odds probability of team 10 beating team 1 at a neutral venue.

(d) First, we build the model matrix.

```
X = matrix(0, nrow=num_games, ncol=num_teams)
for (m in 1:num_games) {
  X[m, 1] = 1
  home = table$Home[m]; if (home != 1) X[m, home] = 1
  away = table$Away[m]; if (away != 1) X[m, away] =-1
  }
```

We run the logistic regression and verify that the coefficients obtained from *optim* and *glm.fit* match.

```
model=glm.fit(X,table$Y,family=binomial())
data.frame(label=c("intercept",teams[-1,1]),
           optim=result$par, glm=model$coefficients)
```

```
## label       optim         glm
## 1                  intercept  0.46268643  0.46258355
## 2           2 Boston Celtics -0.04631213 -0.04648549
## 3            3 Brooklyn Nets -1.61273464 -1.61284430
## 4        4 Charlotte Hornets -0.04467431 -0.04484980
## 5            5 Chicago Bulls -0.34269585 -0.34288850
## 6      6 Cleveland Cavaliers  0.48246253  0.48230358
## 7         7 Dallas Mavericks -0.30639779 -0.30652209
## 8           8 Denver Nuggets -0.84037107 -0.84055558
## 9          9 Detroit Pistons -0.25148051 -0.25161339
## 10  10 Golden State Warriors  1.90274553  1.90258509
## 11        11 Houston Rockets -0.38169132 -0.38186156
## 12         12 Indiana Pacers -0.17399666 -0.17414817
## 13   13 Los Angeles Clippers  0.28898908  0.28888477
## 14     14 Los Angeles Lakers -1.87604922 -1.87617349
## 15      15 Memphis Grizzlies -0.34574217 -0.34587860
## 16            16 Miami Heat -0.01215086 -0.01229379
## 17        17 Milwaukee Bucks -0.82875230 -0.82893701
## 18 18 Minnesota Timberwolves -1.08557680 -1.08567514
## 19   19 New Orleans Pelicans -1.02941242 -1.02954690
## 20        20 New York Knicks -0.92056670 -0.92063507
## 21  21 Oklahoma City Thunder  0.42045913  0.42029832
## 22          22 Orlando Magic -0.75226660 -0.75242422
## 23     23 Philadelphia 76ers -2.55271651 -2.55279786
## 24           24 Phoenix Suns -1.47403237 -1.47425350
## 25 25 Portland Trail Blazers -0.21387094 -0.21399684
## 26       26 Sacramento Kings -0.85175169 -0.85185252
## 27      27 San Antonio Spurs  1.29320880  1.29305390
## 28        28 Toronto Raptors  0.42940629  0.42930221
## 29              29 Utah Jazz -0.45062893 -0.45075953
## 30    30 Washington Wizards -0.39769361 -0.39782372
```