Module 3, Part 1: Generalized Linear Models

BIOS 526

Concepts

- Link function.
- Logistic regression and odds ratio.
- Probit regression.
- Poisson regression.

Readings

- Chapter 3, Wood, S. *Generalized Additive Models*, 2017. Has a nice, self-contained introduction to generalized linear models.

Linear Regression Model

Consider the following multiple linear regression model. For $i = 1, \ldots, n$,

$$y_i = \beta_0 + \sum_{k=1}^{p} \beta_k x_{ik} + \epsilon_i \qquad \epsilon_i \overset{iid}{\sim} N(0, \sigma^2),$$

where x_{ik} is the kth linear predictor for observation i.

The above model assumes

- $\beta_0, \beta_1, \ldots, \beta_p$ are fixed unknown constants;
- only the residual error ϵ_i is random.

Therefore,

1. y_i follows a normal distribution.
2. $E[y_i \,|\, \boldsymbol{x}_i] = \beta_0 + \sum_{k=1}^{p} \beta_k x_{ik}$.

The linear regression part is used to model only the mean function of y_i.

Generalized Linear Regression Model

A generalized linear model (GLM) extends linear regression to other distributions, where the response variable is generated from a distribution in the exponential family.

A GLM involves three ingredients:

1. An exponential family of probability distributions.
2. A linear model $\boldsymbol{x}_i'\boldsymbol{\beta}$.
3. A link function $g()$ and its inverse $g^{-1}()$ relates the linear model to its expectation:

$$E[y_i \,|\, \boldsymbol{x}_i] = \mu_i = g^{-1}(\boldsymbol{x}_i'\boldsymbol{\beta})$$
$$Var[y_i \,|\, \boldsymbol{x}_i] = V(\mu_i) = V(g^{-1}(\boldsymbol{x}_i'\boldsymbol{\beta}))$$

Note: unlike ordinary least squares, the basic form of the GLM does not involve a noise variance (no σ^2).

Exponential Family

The basic form for an exponential family density is

$$f_\theta(y) = \exp\left[\{y\theta - b(\theta)\}/a(\phi) + c(y,\phi)\right],$$

where b, a, and c are known functions, and ϕ is a known scale parameter. There is only one unknown parameter: θ.

In the GLM, θ will be a function of $\boldsymbol{x}_i'\boldsymbol{\beta}$.

Examples of distributions in the exponential family include: normal distribution with **known** variance (link $=$ identity), Bernoulli (logit or probit link), binomial (with fixed number of trials), gamma, exponential (link: negative inverse), and others.

https://en.wikipedia.org/wiki/Generalized_linear_model

Rather than derive expressions for the general case, we will focus on the two most popular models:

1. Binary outcome: $y_i \overset{ind}{\sim}$ Bernoulli (p_i), where p_i is the probability of success.
2. Poisson outcome: $y_i \overset{ind}{\sim}$ Poisson (λ_i), λ_i is the rate parameter, equal to expected number of events.

GLMs

The mean and variance function of y_i can be expressed as a function of the distribution parameters (i.e. p_i for Bernoulli and λ_i for Poisson).

1. Binary outcome:
   $E[y_i] = \mu_i = p_i,$
   $Var[y_i] = V[\mu_i] = V[p_i] = p_i(1 - p_i).$
2. Poisson outcome:
   $E[y_i] = \mu_i = \lambda_i,$
   $Var[y_i] = V[\mu_i] = V[\lambda_i] = \lambda_i.$

A natural approach is to model the mean as a function of linear predictors. A difficulty in modeling non-normal data is that the distributional parameters often have constraints.

1. Binary outcome has expected value $p_i \in (0, 1)$
2. Poisson outcome has expected value $\lambda_i > 0.$

Workspace

Generalized Linear Regression Model

Our solution is to model the transformed mean function:

$$g(\mu_i) = \beta_0 + \sum_{k=1}^{p} \beta_k x_{ik}.$$

The function $g(\cdot)$ is known as the link function.

The link function should have some desirable properties:

1. $g(\cdot)$ should have a range of $(-\infty, \infty)$ because β_k and x_{ik} can take any real value.
2. $g(\cdot)$ should have a domain that corresponds to possible values of μ_i. (i.e. $(0, 1)$ for binary outcome and $(0, \infty)$ for Poisson outcome.
3. $g(\cdot)$ should be 1-to-1. Then,

   $$\mu_i = g^{-1}(\beta_0 + \sum_{k=1}^{p} \beta_k x_{ik}).$$

   A strictly increasing or decreasing function will satisfy this.

GLM for Binary Outcome

For $i = 1, \ldots, n$, assume

$$y_i \overset{ind}{\sim} \text{Bernoulli } (p_i),$$

where $y_i \in \{0, 1\}$, $y_i | p_i$ for $i = 1, \ldots, n$ are independent, and p_i is the probability $y_i = 1$. The probability mass function is given by

$$f(y_i | p_i) = p_i^{y_i} (1 - p_i)^{1 - y_i}.$$

We know that

$$\mu_i = p_i = P(Y_i = 1).$$

So we wish to model

$$g[\, P(Y_i = 1)\,] = \beta_0 + \sum_{k=1}^{p} \beta_k x_{ik}.$$

The two most commonly used link functions are the logistic function and the probit function.

Logistic Regression

The logistic regression is formulated as follows. For $i = 1, \ldots, n$,

$$y_i \overset{ind}{\sim} \text{Bernoulli } (p_i)$$

$$\log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \sum_{k=1}^{p} \beta_k x_{ik}.$$

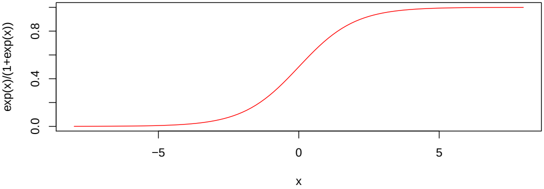
- $g(\mu_i)$ is the log odds of success probability.
- $\log\left(\frac{p_i}{1-p_i}\right) \to -\infty$ when $p_i \to 0$; and $\log\left(\frac{p_i}{1-p_i}\right) \to \infty$ when $p_i \to 1$
- $\log\left(\frac{p_i}{1-p_i}\right)$ is strictly increasing.

Likelihood function: logistic regression

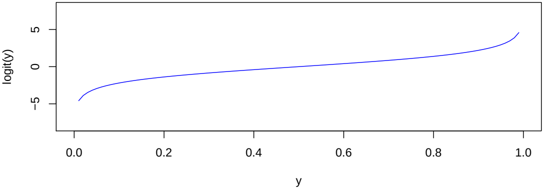
$$\begin{aligned}
\ell(\boldsymbol{\beta}; \mathbf{y}, \boldsymbol{x}) &= \log \prod_{i=1}^{n} p_i^{y_i} (1 - p_i)^{1 - y_i} \\
&= \sum_{i=1}^{n} y_i \log \left(\frac{e^{\boldsymbol{\beta}' \boldsymbol{x}_i}}{1 + e^{\boldsymbol{\beta}' \boldsymbol{x}_i}} \right) + (1 - y_i) \log \left(\frac{1}{1 + e^{\boldsymbol{\beta}' \boldsymbol{x}_i}} \right) \\
&= \sum_{i=1}^{n} y_i \boldsymbol{\beta}' \boldsymbol{x}_i - \log(1 + \exp(\boldsymbol{\beta}' \boldsymbol{x}_i))
\end{aligned}$$

In practice, GLMs are estimated using iteratively reweighted least squares. We won't go into details, but see p. 107 in Wood for more info.

The logistic function:
$p(g) = \exp(g)/(1+\exp(g)) = 1/(1+\exp(-g)),\ (-\infty,\infty) \rightarrow (0,1).$



The logit function: $g(p) = \log(p/(1-p)),\ (0,1) \rightarrow (-\infty,\infty).$



Logistic Regression: Interpretation

Consider a simple logistic model with only one predictor:

$$y_i \overset{ind}{\sim} \text{Bernoulli } (p_i)$$

$$\log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 x_i.$$

When $x_i = 0$:

- $\log\left(\frac{p_i}{1-p_i}\right) = \beta_0.$
- β_0 is interpreted as the baseline log odds.
  Function of the probability of success at baseline:

$$p_i = \frac{e^{\beta_0}}{1+e^{\beta_0}}.$$

- Note that the above function satisfies
  - $p_i \in (0,1)$ for $\beta_0 \in \mathbb{R}$.
  - p_i a strictly increasing function of β_0 .

Logistic Regression: Log odds and log odds ratio

Now we consider the effect of a unit change in x_i:

$$\log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 x_i \quad \text{versus} \quad \log\left(\frac{p_i^*}{1-p_i^*}\right) = \beta_0 + \beta_1 (x_i + 1).$$

Then,

$$\begin{aligned} \log\left(\frac{p_i^*}{1-p_i^*}\right) - \log\left(\frac{p_i}{1-p_i}\right) &= \beta_0 + \beta_1 (x_i + 1) - (\beta_0 + \beta_1 x_i) \\ &= \beta_1. \end{aligned}$$

In words: β_1 is the change in log odds per unit change in x_i.

Equivalently, it is the log odds ratio per unit change in x_i:

$$\beta_1 = \log\left[\frac{p_i^*/(1-p_i^*)}{p_i/(1-p_i)}\right].$$

Logistic Regression: Odds ratio

e^{β_1} is the odds ratio:

$$e^{\beta_1} = \frac{p_i^*/(1-p_i^*)}{p_i/(1-p_i)}.$$

Odds ratio interpretation helpful for indicator variables. Let $x_i = 1$ in exposed group, $x_i = 0$ in unexposed group. Then:

$$\begin{aligned} e^{\beta_1} &= \frac{\left\{\frac{P(y_i=1|x_i=1)}{P(y_i=0|x_i=1)}\right\}}{\left\{\frac{P(y_i=1|x_i=0)}{P(y_i=0|x_i=0)}\right\}} \\ &= \text{odds(Exposed)}/\text{odds(Unexposed)}. \end{aligned}$$

Logistic Regression: Background

- Logistic regression is commonly used because the slope coefficient corresponds to the log odds ratio (OR), a commonly used measure in epidemiology.

- OR is different from relative risk.
- Risk ratio (RR) is $P(y_i = 1|x_i = 1)/P(y_i = 1|x_i = 0)$
- OR is close to RR when the event $y_i = 1$ is rare, but in general, you need a different model to estimate RR.
- OR can be used in retrospective and observational studies.

Logistic Regression: Probability

The effect of a unit change in x_i depends on x_i on the probability scale.

E.g., $\beta_0 = 0$ and $\beta_1 = 2$:

$$\frac{e^{2(0+1)}}{1+e^{2(0+1)}} - \frac{e^{2(0)}}{1+e^{2(0)}} \neq \frac{e^{2(1+1)}}{1+e^{2(1+1)}} - \frac{e^{2(1)}}{1+e^{2(1)}}$$

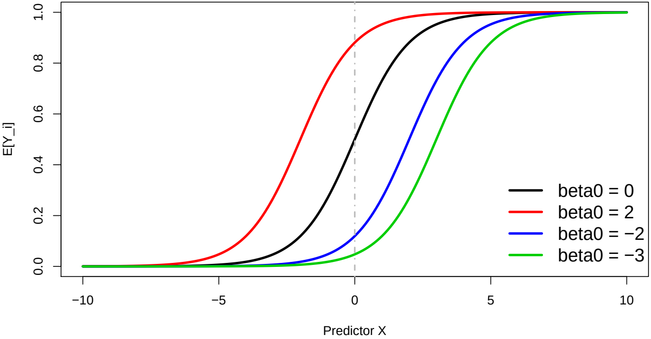
A change in x_i from 0 to 1 increases the probability by 0.38, but a change in x_i from 1 to 2 increases the probability by 0.10.

Intuitively, this has to be the case in order for the probability to max out at 1: $\frac{e^{2(100+1)}}{1+e^{2(100+1)}} - \frac{e^{2(100)}}{1+e^{2(100)}} \approx 1 - 1$.

Logistic Regression: Effects of Baseline Odds

$$\text{logit}(p_i) = \beta_0 + x_i.$$

$$\mu_i = P(Y_i = 1) = \frac{e^{\beta_0 + x_i}}{1 + e^{\beta_0 + x_i}}.$$



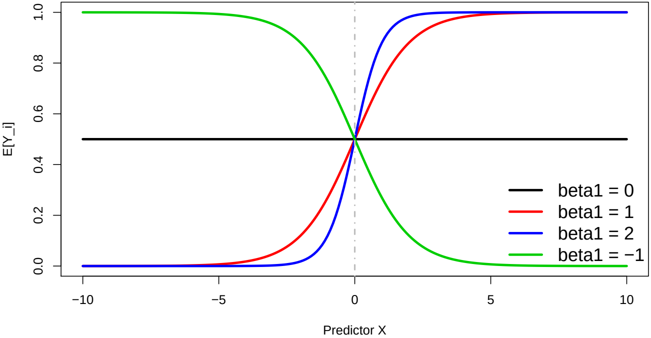
Note:

- The shape is maintained.
- The baseline (intercept) probability changes.

Logistic Regression: Effects of Slope

$$\text{logit}(\mu_i) = \beta_1 x_i.$$

$$\mu_i = P(Y_i = 1) = \frac{e^{\beta_1 x_i}}{1 + e^{\beta_1 x_i}}.$$



Note:

- How the steepness changes.
- How the direction of effect changes.

Logistic Regression: Interpretation

In multiple logistic regression, for $i = 1, \ldots, n$,

$$y_i \overset{ind}{\sim} \text{Bernoulli } (p_i) = \text{Bernoulli } (\mu_i)$$

$$\log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \sum_{k=1}^{p} \beta_k x_{ik}.$$

- β_0 is the log odds when all covariate values equal zero.
- β_k is the log odds ratio associated with covariate k while controlling for other covariates.

Logistic Regression: Interpretation

The estimated (predicted) value is given by

$$\mu_i = p_i = \frac{e^{\beta_0 + \sum_{k=1}^{p} \beta_k x_{ik}}}{1 + e^{\beta_0 + \sum_{k=1}^{p} \beta_k x_{ik}}}$$

Again, the above function is non-linear in x_k, in contrast with normal regression

Inference: Likelihood Ratio Tests

To conduct inference, we appeal to asymptotic results that hold for large-ish n. There are two approaches:

1) Likelihood Ratio Tests (Difference in Deviance)

Let $\boldsymbol{\beta}_F$ be a vector of coefficients of interest. Then to test $H_0 : \boldsymbol{\beta}_F = 0$, we create a full and reduced model. Let $\ell(\hat{\boldsymbol{\beta}}_{\text{Full}})$ be the log-likelihood of the full model, and $\ell(\hat{\boldsymbol{\beta}}_{\text{Reduced}})$ be the LL for the reduced model. Then we reject H_0 if

$$-2\left\{\ell(\hat{\boldsymbol{\beta}}_{\text{Reduced}}) - \ell(\hat{\boldsymbol{\beta}}_{\text{Full}})\right\} > \chi^2_{\nu,1-\alpha}$$

where ν is the difference in the number of parameters between the full and reduced, and $\chi^2_{\nu,1-\alpha}$ is the critical value from a chi-squared distribution with ν degrees of freedom.

Inference: Wald Tests

2) Wald Tests. Under regularity conditions, asymptotically,

$$\begin{aligned}\hat{\boldsymbol{\beta}} &\sim N(\boldsymbol{\beta}, I(\boldsymbol{\beta})^{-1}),\\ I(\boldsymbol{\beta}) &= E\left\{\left(\frac{\partial \ell}{\partial \boldsymbol{\beta}}\right)\left(\frac{\partial \ell}{\partial \boldsymbol{\beta}}\right)'\right\}\\ &= -E\frac{\partial^2 \ell}{\partial \boldsymbol{\beta}\partial \boldsymbol{\beta}'},\end{aligned}$$

where the Hessian $I(\boldsymbol{\beta})$ is called the Fisher information matrix. See Wood p. 106 for details of the expected Hessian which is calculated during iteratively re-weighted least squares.

We can write $\hat{\boldsymbol{\beta}} \sim N(\boldsymbol{\beta}, (\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}\phi)$ where $\mathbf{W}$ contains the "Fisher weights" and $\phi = 1$ in the usual (not overdispersed) GLM.

In R, the default `summary(glmmodel)` is a Wald-type test: $\hat{\beta}_j / se(\hat{\beta}_j)$, where $se(\hat{\beta}_j)$ is extracted from the square root of the jth diagonal of the above covariance.

Binary Outcome Example

Dataset: a cohort of live births from Georgia born in the year 2001 (N = 77,340).

Variables:

- ptb: indicator for whether the baby from pregnancy i was born preterm ($<$ 37 weeks).
- age: the mother's age at delivery (centered at age 25).
- $male$: indicator of the baby's sex (1 = male; 0=female).
- $tobacco$: indicator for mother's tobacco use during pregnancy (1 = yes; 0 = no)

The *glm ( )* Function

Fitting a GLM model in R is very similar to a linear regression model. We need to specify the distribution (binomial) and the link function (logit).

```
glm(formula = ptb ~ age + male + tobacco, family = binomial(link = "logit"),
    data = dat)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.5160  -0.4236  -0.4103  -0.4088   2.2500

Coefficients:
              Estimate Std. Error  z value Pr(>|z|)
(Intercept) -2.4370033  0.0200791 -121.370  < 2e-16 ***
age         -0.0006295  0.0021596   -0.291  0.77068
maleM        0.0723659  0.0258672    2.798  0.00515 **
tobacco      0.4096495  0.0534627    7.662 1.83e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Birth Outcome Analysis

- Preterm delivery was significantly associated with male babies (p-value$= 0.005$) when controlling for age and mother's smoking status. The odds ratio of a preterm birth for a male baby versus a female baby was 1.07 (95% CI: 1.02, 1.13).
  OR $= e^{0.0723} = 1.07$
  CI $= e^{(0.0723 \pm 1.96 * 0.0258)} = (1.02, 1.13)$
- Note: transform the intervals. Do NOT transform standard errors (requires delta method).
- Preterm delivery was significantly associated with whether the mother smoked during pregnancy ($p < 0.001$) when controlling for age and the baby's sex. The odds ratio for mother's that smoked versus did not smoke was $e^{0.409} = 1.51$ (95% CI: 1.36, 1.67).

Birth outcome analysis, cont.

- The baseline proportion (female babies born to mother of age 25 who didn't smoke) of preterm delivery was

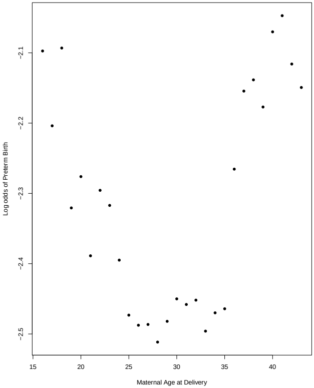
$$\frac{e^{-2.437}}{1+e^{-2.437}} = 0.080.$$

- We didn't find an effect of mother's age.

Birth Outcome Analysis - Mother's Age

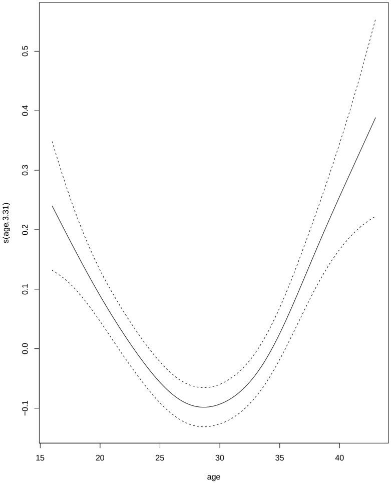
We assumed that the mother's age has a linear effect on log odds of preterm birth. Is this a reasonable assumption?

Explore this by calculating % preterm births for each age group.



Generalized Additive Model

In Module 5, we will model this non-linearly using splines:



P-values using LRTs

```
> library(car)
> # LRTs:
> Anova(fit)
Analysis of Deviance Table (Type II tests)

Response: ptb
        LR Chisq Df Pr(>Chisq)
age        0.085  1   0.770669
male       7.834  1   0.005126 **
tobacco   53.648  1  2.399e-13 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Probit Regression

Probit regression is an alternative approach to model binary data. It still assumes the Bernoulli model, but uses a different link function. For $i = 1, \ldots, n$,

$$y_i \overset{ind}{\sim} \text{Bernoulli }(p_i)$$

$$\Phi^{-1}(p_i) = \beta_0 + \sum_{k=1}^{p} \beta_k x_{ik},$$

where Φ^{-1} is the inverse cumulative distribution function of a standard normal distribution. Recall $\Phi^{-1}(x)$ asks what Z-value gives a cumulative probability of x?

- Example: $\Phi^{-1}(0.5) = 0$ and $\Phi^{-1}(0.975) = 1.96$.
- Note $\Phi^{-1}(p_i)$ is strictly increasing, has range $(-\infty, \infty)$ and domain $(0, 1)$.

Probit Regression

The probit link function results in

$$p_i = \Phi\big(\beta_0 + \sum_{k=1}^{p} \beta_k x_{ik}\big).$$

Therefore, $P(y_i = 1)$ is viewed as the probability of a standard normal variable being less than $\beta_0 + \sum_{k=1}^{p} \beta_k x_{ik}$.

Probit regression has a very attractive latent variable (i.e., unobserved) interpretation. Let Z_i denote a latent variable associated with each binary outcome.

$$Z_i \overset{ind}{\sim} N(\beta_0 + \sum_{k=1}^{p} \beta_k x_{ik},\, 1).$$

Then

$$\begin{aligned} P(Z_i > 0) = 1 - P(Z_i < 0) &= 1 - P\left(\frac{0 - (\beta_0 + \sum_{k=1}^{p} \beta_k x_{ik})}{1} < 0\right) \\ &= 1 - \Phi\big(-(\beta_0 + \sum_{k=1}^{p} \beta_k x_{ik})\big) = \Phi(\beta_0 + \sum_{k=1}^{p} \beta_k x_{ik}). \end{aligned}$$

Probit Regression: Latent Variable Representation

We can rewrite

$$y_i \overset{ind}{\sim} \text{Bernoulli } (p_i) \qquad \Phi^{-1}(p_i) = \beta_0 + \sum_{k=1}^{p} \beta_k x_{ik}$$

as a hierarchical model:

$$1.\ \ Z_i \overset{ind}{\sim} N\big(\beta_0 + \sum_{k=1}^{p} \beta_k x_{ik}, 1\big)$$

$$2.\ \ y_i = \begin{cases} 0 & \text{if } Z_i < 0 \\ 1 & \text{if } Z_i > 0 \end{cases}$$

Therefore we assume the binary outcome $y_i = 1$ when its latent variable Z_i passes the threshold 0.

The probability of this occurring depends on the mean of the latent variable Z_i. Larger mean $(\beta_0 + \sum_{k=1}^{p} \beta_k x_{ik})$ increases the the probability of $y_i = 1$.

Probit Regression: Interpretation

Consider a simple probit model with only one predictor:

$$y_i \overset{ind}{\sim} \text{Bernoulli } (p_i)$$
$$p_i = \Phi(\beta_0 + \beta_1 x_i).$$

Interpretation of the regression coefficients is arguably more challenging. Represents change in z-score.

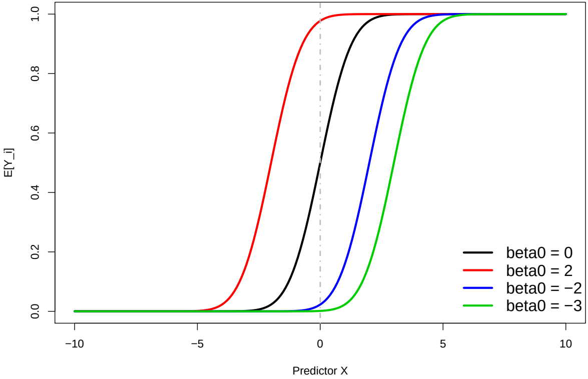
- The baseline probability is $\Phi(\beta_0)$.
- The effect of a unit increase in x_i on $P(y_i = 1)$ is

$$\Phi(\beta_0 + \beta_1 x_i + \beta_1) - \Phi(\beta_0 + \beta_1 x_i),$$

  which again depends on the value of x_i.

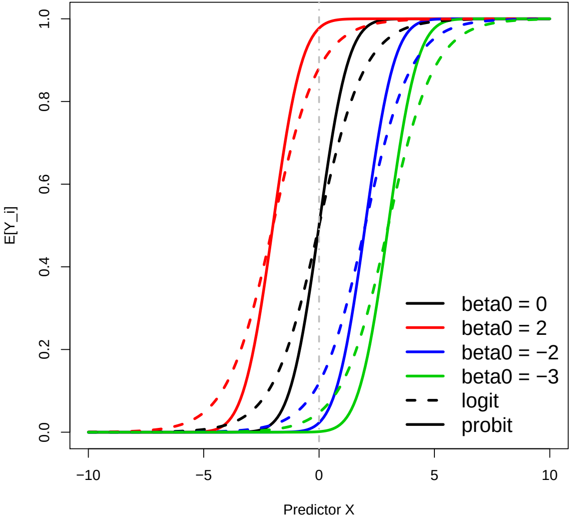
Probit Regression: Effects of Intercept

$$p_i = \mu_i = \Phi(\beta_0 + \beta_1 x_i).$$



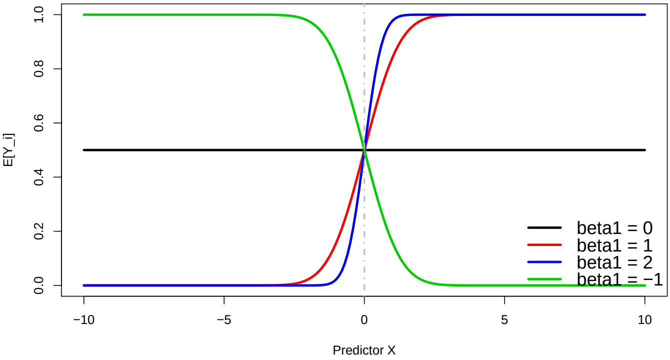
Very similar behaviors as logistic regression. Slightly different tail behaviors compared to a logit link function.

Logit and Probit Regression: Effects of Intercept



Probit Regression: Effects of Slope

$$p_i = \mu_i = \Phi(\beta_1 x_i).$$



The *glm ( )* Function with probit

```
glm(formula = ptb ~ age + male + tobacco, family = binomial(link = "probit"),
    data = dat)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.5158  -0.4237  -0.4104  -0.4087   2.2509

Coefficients:
              Estimate Std. Error  z value Pr(>|z|)
(Intercept) -1.4024285  0.0099738 -140.612  < 2e-16 ***
age         -0.0003746  0.0010793   -0.347   0.7285
maleM        0.0363566  0.0129215    2.814   0.0049 **
tobacco      0.2102264  0.0281156    7.477 7.59e-14 ***
```

Comparing the logistic and probit regression model, we note the regression coefficients are qualitatively similar but the magnitude differs.

Poisson Regression

A Poisson regression is specified as follows. For $i = 1, \ldots, n$,

$$y_i \overset{ind}{\sim} \text{Poisson } (\lambda_i)$$

$$\log(\lambda_i) = \beta_0 + \sum_{k=1}^{p} \beta_k x_{ik}.$$

For a Poisson distributed random variable,

$$E\ y_i = \lambda_i,$$
$$V\ y_i = \lambda_i$$

Poisson regression is often used to model count data. Examples include daily mortality in a city, number of HIV infected individuals in a neighborhood, and number of medical errors at a hospital.

- Here the link function is $\log(\cdot)$.
- $\log(\cdot)$ has domain $(0, \infty)$ and range $(-\infty, \infty)$, and is strictly increasing.

Poisson Regression Interpretation

Consider a simple Poisson regression model with only one covariate:

$$y_i \overset{ind}{\sim} \text{Poisson}\ (\lambda_i)$$

$$\log(\lambda_i) = \beta_0 + \beta_1 x_i.$$

When $x_i = 0$, $\log(\lambda_i) = \beta_0$.

- $e^{\beta_0} = \lambda_i$ is the baseline expected counts.

Poisson Regression Interpretation

Now consider a unit change in x

$$\log(\lambda_i) = \beta_0 + \beta_1 x_i \qquad \log(\lambda_i^*) = \beta_0 + \beta_1(x_i + 1).$$

Note that

$$\beta_1 = \log(\lambda_i^*) - \log(\lambda_i)$$

- $e^{\beta_1} = \lambda_i^* / \lambda_i$ is the relative change (rate) in count per unit change in x.
- For continuous variables with $\beta_1 > 0$, the rate increases by $100 * (e^{\beta_1} - 1)$% for every unit increase in x.
- For factors with $\beta_1 > 0$, the rate increases by $100 * (e^{\beta_j} - 1)$% for level j relative to baseline.

Covariate impacts are multiplicative rather than additive (applies to log models in general) on the count scale:

$$E\ y_i = e^{\beta_0} e^{\beta_1 x_{i1}} \cdots e^{\beta_p x_{ip}}$$

Example: bacteria counts

Dataset: antibiotic resistance in a mutation of *E. coli*.

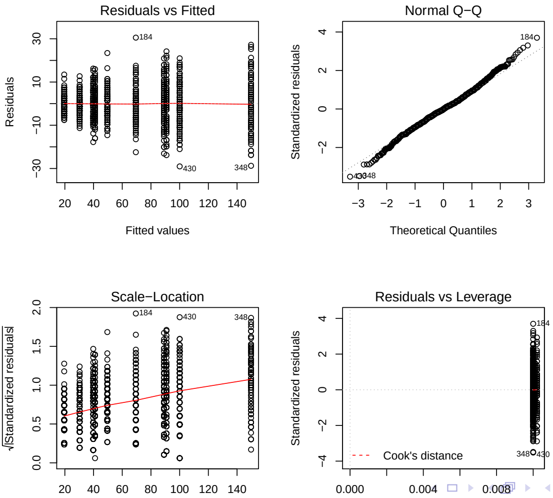
Variables:

- *Colony*: the number of ampicillin-resistant mutant colonies
- *Conc*: the concentration of novobiocin (μg/ml)
- *Media*: the type of media used for bacterial growth.

The experiment involved two media preparations (LB and M9), 5 concentrations of novobiocin, and 100 replicates for each media-concentration combination. TNTC (too numerous to count) were recorded when the number of colonies exceeded 300.

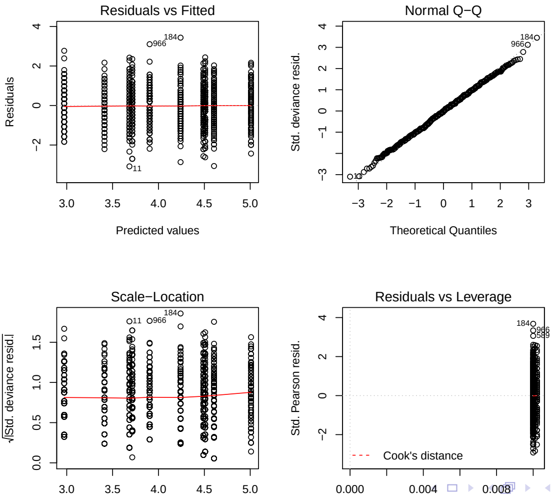
Modeling count data

```
lm_colony = lm(Colony_numeric~factor(Conc)*Media,data=colonydata)
```



Modeling count data: Poisson

```
glm_colony = glm(Colony_numeric~factor(Conc)*Media,data=colonydata,family = "poisson")
```



Residuals in glms

The usual plot of residual versus fitted is not useful in GLMs because of the relationship between the mean and variance.

The deviance residuals have an approximate normal distribution.

The deviance residual is

$$\hat{\epsilon}_i^d = \mathsf{sign}(y_i - \hat{\mu}_i)\sqrt{d_i}.$$

where d_i is the ith term in the calculation of the deviance. For details, see p.113 in Wood 2017. (This plot is not useful for 0/1 data as common in logistic regression.)

Goodness of fit tests, quasipoisson

In glms, the deviance performs a role similar to the sum of squared errors in OLS:

$$D(\hat{\boldsymbol{\beta}}) = 2\left\{\ell(\hat{\boldsymbol{\beta}}_{\text{max}}; \mathbf{y}) - \ell(\hat{\beta}; \mathbf{y})\right\}$$

where $\ell(\hat{\boldsymbol{\beta}}_{\text{max}}; \mathbf{y})$ is the "saturated model," equal to likelihood evaluated at $\hat{\mu}_i = y_i$.

Asymptotically, $D(\hat{\boldsymbol{\beta}}) \sim \chi^2_{n-p}$

One can perform a deviance test to examine goodness of fit. The null hypothesis is that the model fits the data. $p < 0.05$ indicates a problem (i.e., lack of fit).

```
with(glm_colony, cbind(res.deviance = deviance, df = df.residual,
p = pchisq(deviance, df.residual, lower.tail=FALSE)))

res.deviance  df         p
[1,]     973.8841 987 0.6108368
```

Here, $p > 0.05$, from which we conclude that the model provides an adequate fit.

Overdispersion

In Poisson, the assumption that $Ey_i = Vy_i$ is often violated.

One can add an additional overdispersion parameter, also called a scale parameter.

One can adjust the parameter variances by the scale parameter:

$$\hat{\boldsymbol{\beta}} \sim N(0, I(\boldsymbol{\beta})^{-1}\phi)$$

We will see this again in GEEs.

Section 3.1.5 in Wood describes three different estimators of ϕ.

Quasipoisson in GLM

```
> glm_colony_quasi = glm(Colony_numeric~factor(Conc)*Media,data=colonydata,family = "quasi
> summary(glm_colony_quasi)

Call:
glm(formula = Colony_numeric ~ factor(Conc) * Media, family = "quasipoisson",
    data = colonydata)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.0815  -0.7615  -0.0047   0.6573   3.4344

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)
(Intercept)              3.683308   0.015807 233.023  < 2e-16 ***
factor(Conc)100          0.557587   0.019786  28.181  < 2e-16 ***
factor(Conc)200          0.806339   0.018981  42.481  < 2e-16 ***
factor(Conc)250          1.325726   0.017764  74.631  < 2e-16 ***
factor(Conc)300          0.922162   0.018660  49.418  < 2e-16 ***
MediaM9                 -0.710845   0.027448 -25.898  < 2e-16 ***
factor(Conc)100:MediaM9 -0.118573   0.034923  -3.395 0.000713 ***
factor(Conc)200:MediaM9 -0.064499   0.033221  -1.942 0.052480 .
factor(Conc)250:MediaM9  0.210650   0.030490   6.909 8.75e-12 ***
factor(Conc)300:MediaM9  0.009164   0.032406   0.283 0.777404
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for quasipoisson family taken to be 0.983908)
```

Back to original model

When the data are slightly underdispersed, i.e., dispersion parameter<1, and there is no evidence of lack of fit, I suggest using the original model:

```
     > summary(glm_colony)

Call:
glm(formula = Colony_numeric ~ factor(Conc) * Media, family = "poisson",
    data = colonydata)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.0815  -0.7615  -0.0047   0.6573   3.4344

Coefficients:
                         Estimate Std. Error z value Pr(>|z|)
(Intercept)              3.683308   0.015935 231.140  < 2e-16 ***
factor(Conc)100          0.557587   0.019947  27.953  < 2e-16 ***
factor(Conc)200          0.806339   0.019136  42.138  < 2e-16 ***
factor(Conc)250          1.325726   0.017908  74.028  < 2e-16 ***
factor(Conc)300          0.922162   0.018812  49.019  < 2e-16 ***
MediaM9                 -0.710845   0.027671 -25.689  < 2e-16 ***
factor(Conc)100:MediaM9 -0.118573   0.035208  -3.368 0.000758 ***
factor(Conc)200:MediaM9 -0.064499   0.033491  -1.926 0.054126 .
factor(Conc)250:MediaM9  0.210650   0.030738   6.853 7.23e-12 ***
factor(Conc)300:MediaM9  0.009164   0.032670   0.280 0.779097
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Interpretation

- The rate here is colonies per petri dish.
- Intercept: the log of the expected number of colonies is 3.68 in LB media with no novobiocin. Equivalently, the log rate is 3.68 in LB media with no novobiocin.
- The estimated number of colonies and 95% CI for this baseline is $e^{3.68}$, $e^{3.68-1.96*0.016} - e^{3.68+1.96(0.016)} =$ 39.6 (38.4 - 41).
- The relative rate in Media M9 with concentration of novobiocin equal to 300 is $e^{0.009}$, i.e., the rate increases by 0.9%, which is not significant ($p > 0.05$).

Example with overdispersion

For educational purposes, consider this poor fitting model:

```
> glm_nomedia_quasi = glm(Colony_numeric~factor(Conc),data=colonydata,
family = "quasipoisson")
> summary(glm_nomedia_quasi)
Call:
glm(formula = Colony_numeric ~ factor(Conc), family = "quasipoisson",
    data = colonydata)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-5.6832  -2.8772  -0.2983   2.5557   6.2330

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)      3.38805    0.03727   90.90   <2e-16 ***
factor(Conc)100  0.52177    0.04701   11.10   <2e-16 ***
factor(Conc)200  0.78726    0.04493   17.52   <2e-16 ***
factor(Conc)250  1.40432    0.04162   33.74   <2e-16 ***
factor(Conc)300  0.92691    0.04400   21.06   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for quasipoisson family taken to be 8.185913)

    Null deviance: 21484.0  on 996  degrees of freedom
Residual deviance:  8249.4  on 992  degrees of freedom
AIC: NA
```

GOF test

```
 > glm_nomedia = glm(Colony_numeric~factor(Conc),data=colonydata,family = "poisson")
> summary(glm_nomedia)

Call:
glm(formula = Colony_numeric ~ factor(Conc), family = "poisson",
    data = colonydata)
...
(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 21484.0  on 996  degrees of freedom
Residual deviance:  8249.4  on 992  degrees of freedom
AIC: 14127

> # versus:
> with(glm_nomedia, cbind(res.deviance = deviance, df = df.residual, p = pchisq(deviance, 
     res.deviance  df p
[1,]     8249.368 992 0
```