Week 2 Notes: Basics of Generalized Linear Models

This week, we will discuss *generalized linear models*. Much of the exposition here is derived from the textbook *Generalized Linear Models* by McCullaugh and Nelder (1989). I am operating under the assumption that your previous classes have covered Frequentist and Bayesian approaches to linear models in sufficient detail, and that you have a basic understanding of how maximum likelihood estimation works and that the MLEs are asymptotically normal; if you don't know those things, we will discuss asymptotics of MLE in a later lecture, but it would be advisable to read up on ML estimation in the meantime as in these notes: https://www.math.arizona.edu/~jwatkins/o-mle.pdf.

1 Motivation

Generalized linear models were introduced to resolve many of the limitations that arise from linear models — perhaps most importantly, the heteroskedasticity that arises naturally from Poisson and binomial/Bernoulli response models.

In the beforetimes, when there was software to fit linear models but not generalized linear models, folks used a variety of hacks to deal with the fact that various types of data are intrinsically heteroskedsatic. For example, if Y is a count we generally expect that $\text{Var}(Y) \geq \mathbb{E}(Y)$ (for Poisson data, there is equality). One approach is to transform the data to be homoskedastic, i.e., we could use the model

$$g(Y_i) = X_i^\top \beta + \epsilon_i$$

for some transformation $g(\cdot)$, with $\mathbb{E}(\epsilon_i) = 0$ and $\text{Var}(\epsilon_i) = \sigma^2$. Usually, we would take $g(y)$ to be a *variance stabilizing transformation*.

Exercise 1: Variance Stabilizing Transformations

Suppose that $Y \sim \text{Poisson}(\lambda)$. Using the expansion

$$g(y) \approx g(\lambda) + (y - \lambda)\, g'(\lambda)$$

find a transformation $g(\cdot)$ such that the variance of $g(Y)$ is approximately constant.

Exercise 2: Limitations

Explain some of the deficiencies of the model from the previous exercise. For example: is there any issues with the interpretation of β when compared with the usual linear regression model?

The framework of generalized linear models allows for the same ideas underlying linear models to be extended to other response types (count, discrete, non-negative, etc) without resorting to the contortions of Exercise 1.

2 Generalized Linear Models

The class of *generalized linear models* assumes that we are working with a dependent variable Y_i that has a distribution in an *exponential dispersion family*.

Definition 1: Exponential Dispersion Family

A family of distributions $\{f(\cdot;\theta,\phi) : \theta \in \Theta, \phi \in \Phi\}$ is an *exponential dispersion family* if we can write

$$f(y;\theta,\phi) = \exp\left\{\frac{y\theta - b(\theta)}{\phi} + c(y;\phi)\right\},$$

for some *known* functions $b(\cdot)$ and $c(\cdot,\cdot)$. The parameter θ is referred to as the *canonical parameter* of the family and ϕ is referred to as the *dispersion parameter*.

Exercise 3: Examples of Exponential Dispersion Families

Show that the following families are types of exponential dispersion families, and find the corresponding b, c, θ, ϕ.

1. $Y \sim \text{Normal}(\mu, \sigma^2)$
2. $Y = Z/N$ where $Z \sim \text{Binomial}(N, p)$
3. $Y \sim \text{Poisson}(\lambda)$
4. $Y \sim \text{Gam}(\alpha, \beta)$ (parameterized so that $\mathbb{E}(Y) = \alpha/\beta$).

Using this definition, we can define the class of generalized linear models. Generalized linear models serve the role of generalizing the normal linear regression model $Y_i = X_i^\top \beta + \epsilon_i$ to allow for Y_i to discrete (or otherwise non-normally-distributed).

Definition 2: Generalized Linear Models

Suppose that we have $\mathcal{D} = \{(Y_i, x_i) : i = 1, \ldots, N\}$ (with the x_i's regarded as fixed constants). We say that the Y_i's follow a *generalized linear model* if:

1. Y_i has density/mass function

$$f(y_i \mid \theta_i, \phi/\omega_i) = \exp\left\{\frac{y_i\,\theta_i - b(\theta_i)}{\phi/\omega_i} + c(y_i; \phi/\omega_i)\right\}$$

   where the coefficients $\omega_1, \ldots, \omega_N$ are known. This is referred to as the *stochastic component* of the model.

2. For some known (invertible) *link function* $g(\mu)$ we have

$$g(\mu_i) = x_i^\top \beta$$

   where $\mu_i = \mathbb{E}(Y_i \mid \theta_i, \phi/\omega_i)$. This is referred to as the *systematic component* of the model. The term $\eta_i = x_i^\top \beta$ is known as the *linear predictor*.

The reason we allow for the inclusion of individual-specific weights ω_i's is to allow for us to model (for example) binomial-type data where the sample sizes for the different units are different.

Exercise 4: GLM Moments

Suppose that $Y \sim f(y; \theta, \phi/\omega)$ for some exponential dispersion family. Show that

1. $\mathbb{E}(Y \mid \theta, \phi/\omega) = b'(\theta)$; and
2. $\mathrm{Var}(Y \mid \theta, \phi/\omega) = \frac{\phi}{\omega} b''(\theta)$.

Hint: The log-likelihood is given by

$$\log f = \frac{y\theta - b(\theta)}{\phi/\omega} + c(y, \phi/\omega).$$

Use the *score equations*

$$\begin{aligned} \mathbb{E}\{s(y; \theta, \phi/\omega) \mid \theta, \phi/\omega\} &= \mathbf{0} \qquad \text{and} \\ \mathrm{Var}\{s(y; \theta, \phi/\omega) \mid \theta, \phi/\omega\} &= -\mathbb{E}\{\dot{s}(y; \theta, \phi/\omega)\}, \end{aligned}$$

to derive the result, where

$$s(y; \theta, \phi/\omega) = \frac{\partial}{\partial \theta} \log f \qquad \text{and} \quad \dot{s}(y; \theta, \phi/\omega) = \frac{\partial^2}{\partial \theta \partial \theta^\top} \log f$$

are the gradient and Hessian matrix of $\log f$ with respect to θ.

From Exercise 4 we immediately have

$$\theta_i = (b')^{-1}(\mu_i) = (b')^{-1}\{g^{-1}(x_i^\top \beta)\}$$

provided that b' and g both have an inverse. Note that GLMs are *heteroskedastic models*, as $\text{Var}(Y \mid \theta, \phi/\omega)$ depends on $\mathbb{E}(Y_i \mid \theta, \phi/\omega)$. In particular, we have

$$\text{Var}(Y \mid \theta, \phi/\omega) = \frac{\phi}{\omega} b''(\theta) = \frac{\phi}{\omega} b''\{(b')^{-1}(\mu)\} = \frac{\phi}{\omega} V(\mu).$$

The function $V(\mu) = b''\{(b')^{-1}(\mu)\}$ is sometimes called the *variance function* of the GLM.

Exercise 5: Variance Functions

Show the following.

a. For the Poisson regression model, $V(\mu) = \mu$.

b. For the binomial proportion regression model, $V(\mu) = \mu(1 - \mu)$.

Exercise 6: Existence of Necessary Inverse

Argue (informally) that there exists an inverse function for $b'(\theta)$ provided that $\text{Var}(Y \mid \theta, \phi) > 0$ for all (θ, ϕ).

Exercise 7: Sanity Check

To convince yourself of the correctness of Exercise 4, use the results to compute the mean and variance of the Normal(μ, σ^2) and Gam(α, β) distributions.

Exercise 8: Canonical Link Function

To specify a GLM we must choose the so-called *link function* $g(\mu)$. A convenient choice (for reasons we will discuss later) is $g(\mu) = (b')^{-1}(\mu)$. This is known as the *canonical link*. By definition this gives the model

$$f(y_i \mid x_i, \omega_i, \theta, \phi) = \exp\left\{ \frac{y_i x_i^\top \beta - b(x_i^\top \beta)}{\phi/\omega_i} + c(y_i; \phi/\omega_i) \right\},$$

i.e., we use the exponential dispersion family with $\theta_i = x_i^\top \beta$.

a. Show $Y \sim \text{Normal}(\mu, \sigma^2)$ has the identity as the canonical link $g(\mu) = \mu$.

b. Show $Y \sim \text{Poisson}(\lambda)$ has the log-link as the canonical link $g(\mu) = \log \mu$.

c. Show that $Y = Z/n$ with $Z \sim \text{Binomial}(n, p)$ has the logit link as the canonical link $g(\mu) = \log\{\mu/(1 - \mu)\}$.

d. Show that $Y \sim \text{Gam}(\alpha, \beta)$ has the inverse as the canonical link $g(\mu) = -1/\mu$.

e. The canonical link for gamma GLMs (while commonly used in some fields) is used

> far less than for other types of GLMs, for one very good reason. What is that reason?

3 Fitting GLMs in R

We can fit GLMs in R via maximum likelihood by using the `glm` command; generally, fitting a GLM will look like this

```
my_glm <- glm(
  response ~ predictor_1 + predictor_2 + and_so_forth,
  data = my_data,
  family = my_family
)
```

The `family` argument tells R which type of GLM to fit: we will mostly use `family = binomial` for logistic regression or `family = poisson` for Poisson regression. It is also possible to change the link function with the `family` command; for example, doing `family = binomial("probit")` corresponds to fitting a binomial GLM using the *probit* link $g(\mu) = \Phi^{-1}(\mu)$ where $\Phi(z)$ is the cdf of a standard normal distribution (more on specific settings for link functions later). You can get information on all the options by running `?glm` in the R console.

The easiest way to fit a GLM in the Bayesian paradigm is probably to use the `rstanarm` package in R.

```
install.packages("rstan")
install.packages("rstanarm")
```

After installing the package we can fit GLMs using something like this:

```
my_glm <- rstanarm::stan_glm(
  response ~ predictor_1 + predictor_2 +  and_so_forth,
  data = my_data,
  family = my_family
)
```

The `rstanarm` package will use a "default" prior that places independent normal priors on the β_j's, but this can be changed; see `?rstanarm::stan_glm` for details on the priors that are available. The default priors are designed to give reasonable answers across a wide variety of problems encountered in practice.

4 Logistic Regression

A particular case of a GLM takes

$$Y_i = Z_i/n_i \qquad \text{where} \qquad Z_i \sim \text{Binomial}(n_i, p_i).$$

When used with the canonical link of Exercise 8 we arrive at the *logistic regression model*

$$p_i = \frac{\exp(x_i^\top \beta)}{1 + \exp(x_i^\top \beta)}.$$

Defining $\text{logit}(p) = \log\{p/(1-p)\}$, we equivalently can express the model as

$$\text{logit}(p_i) = x_i^\top \beta.$$

This model is referred to as the *logistic regression model*, and it is used to model outcomes that correspond to counts from a binomial experiment (i.e., repeatedly flipping a coin n_i times with probability p_i of heads). The special case $n_i = 1$ is also common, in which case the outcomes Y_i are *binary*.

What the Coefficients Represent

Exercise 9: Logistic Regression Coefficients

Suppose we fit a logistic regression model $\text{logit}(p_i) = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2}$. Logistic regression models are often interpreted in terms of *odds ratios*; the odds of success of observational unit i relative to i' is given by

$$\frac{\text{Odds}(Y_i = 1 \mid X_i)}{\text{Odds}(Y_{i'} = 1 \mid X_{i'})} = \frac{\Pr(Y_i = 1 \mid X_i)\ \Pr(Y_{i'} = 0 \mid X_{i'})}{\Pr(Y_i = 0 \mid X_i)\ \Pr(Y_{i'} = 1 \mid X_{i'})}$$

Show that, if X_i and $X_{i'}$ are identical except that $X_{i2} = X_{i'2} + \delta$, then the odds ratio is given by $e^{\beta_2 \delta}$. That is *shifting a covariate by δ has a multiplicative effect on the odds of success, inflating the odds by a factor of $e^{\beta_2 \delta}$.*

Bernoulli Regression: The Challenger Shuttle Explosion

On January 28$^{\text{th}}$, 1986, the space shuttle Challenger broke apart just after launch, taking the lives of all seven crew members. This example is taken from an article by Dalal et al. (1989), which examined whether the incident should have been predicted, and hence prevented, on the basis of data from previous flights. The cause of failure was ultimately attributed to the failure of a crucial shuttle component know as the O-rings; these components had been tested prior to the launch to see if they could hold up under a variety of temperatures.

The dataset `Challenger.csv` consists of data from test shuttle flights. This can be loaded using the following commands.

```
library(tidyverse)

f <- str_c("https://raw.githubusercontent.com/theodds/",
           "SDS-383D/main/Challenger.csv")

challenger <- read.csv(f) %>%
  drop_na() %>%
  mutate(Fail = ifelse(Fail == "yes", 1, 0))

head(challenger)
```

```
  FlightNumber Temperature Pressure Fail nFailures Damage
1            1          66       50    0         0      0
2            2          70       50    1         1      4
3            3          69       50    0         0      0
4            5          68       50    0         0      0
5            6          67       50    0         0      0
6            7          72       50    0         0      0
```

Our Goal: The substantive question we are interested in is whether those in charge of the Challenger launch should have known that the launch was dangerous and delayed it until more favorable weather conditions. In fact, engineers working on the shuttle had warned beforehand that the O-rings were more likely to fail at lower temperatures. **Concretely, we are interested in knowing what the probability of an O-ring failure would be if we repeated the Challenger launch under similar conditions.**

Our Model: To help answer our substantive question, we will consider a model for whether an O-Ring failure occurred on a given flight ($Y_i = 1$ if an O-ring failed, $Y_i = 0$ otherwise) given the temperature $\texttt{temp}_i$. The most general model we could use would be $p_i = f(\texttt{temp}_i)$ for some function $f(\cdot)$; there is nothing wrong with this per-se, but it is useful to consider a model with a more interpretable structure

$$\text{logit}(p_i) = \beta_0 + \beta_{\text{temp}} \times \text{temp}_i.$$

In `R` we can fit the this model by maximum likelihood as follows.

```
challenger_fit <- glm(
  Fail ~ Temperature,
  data = challenger,
```

```
  family = binomial
)
```

One way to extract information out of a fitted GLM is to use the `summary` function:

```
summary(challenger_fit)
```

```
Call:
glm(formula = Fail ~ Temperature, family = binomial, data = challenger)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  15.0429     7.3786   2.039   0.0415 *
Temperature  -0.2322     0.1082  -2.145   0.0320 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 28.267  on 22  degrees of freedom
Residual deviance: 20.315  on 21  degrees of freedom
AIC: 24.315

Number of Fisher Scoring iterations: 5
```

The main parts of the output we are interested in are the columns `Estimate` which gives the maximum likelihood estimates of β_0 (`Intercept`) and β_1 (`Temperature`), `Std. Error` which gives an estimate of the standard error of the MLEs, and the last column which gives a P-value for the test that these coefficients are equal to zero. The rest of the output, while important, is not of direct interest to us at the moment.

Familiarize yourself with the functions `coef`, `vcov`, and `confint` in addition to `summary`; what do these functions do? *Note: generally speaking, the P-values in the output of `summary` are not the best ones, and it is better to base inference on likelihood ratio tests, which we will discuss in a future lecture. Similarly, don't get confidence intervals by adding/subtracting two standard errors, use the `confint` function instead, which inverts a likelihood ratio test and will be discussed in detail later.*

Predictions from the logistic regression model can be obtained using the `predict` function:

```r
predict(challenger_fit,
        newdata = data.frame(Temperature = c(40, 50, 60)),
        type = 'response',
        se.fit = TRUE)
```

```
$fit
        1         2         3
0.9968475 0.9687735 0.7527135

$se.fit
          1           2           3
0.009674669 0.061205420 0.190948130

$residual.scale
[1] 1
```

A Bayesian version can also be fit as follows.

```r
challenger_bayes <- rstanarm::stan_glm(
  Fail ~ Temperature,
  data = challenger,
  family = binomial
)
```

Note: please step through these lines of code below on your own to make sure you understand what each line is doing!!! Inspect the objects on your own as well.

Using the Bayesian version, let's plot the samples of the function

$$f(\texttt{temp}) = \{1 + \exp(-\beta_0 - \beta_1 \texttt{temp})\}^{-1}.$$

We select 200 of the 4000 posterior samples at random for display purposes. I highly encourage you to step through this code on your own to understand what each line does.

```r
## For Reproducibility
set.seed(271985)

## Converts the rstanarm object to a matrix
beta_samples <- as.matrix(challenger_bayes)

## Some Colors
pal <- ggthemes::colorblind_pal()(8)
```

```r
## Set up plotting region
plot(
  x = challenger$Temperature,
  y = challenger$Fail,
  ylab = "Failure?",
  xlab = "Temperature",
  type = 'n'
)

## A function for adding estimate
plot_line <- function(beta, col = 'gray') {
  plot(function(x) 1 / (1 + exp(-beta[1] - beta[2] * x)),
       col = col, add = TRUE, xlim = c(40, 90), n = 200)
}

## Apply plot_line for a random collection of betas
tmpf <- function(i) plot_line(beta_samples[i,])
tmp <- sample(1:4000, 200) %>% lapply(tmpf)

## Get the Bayes estimate of the probability
tempgrid <- seq(from = 40, to = 90, length = 200)
bayes_est <- predict(challenger_bayes,
  type = 'response',
  newdata = data.frame(Temperature = tempgrid)
)
lines(tempgrid, bayes_est, col = pal[3], lwd = 4)

## Adding the observations
points(
  x = challenger$Temperature,
  y = challenger$Fail,
  pch = 20,
  col = pal[4]
)
```

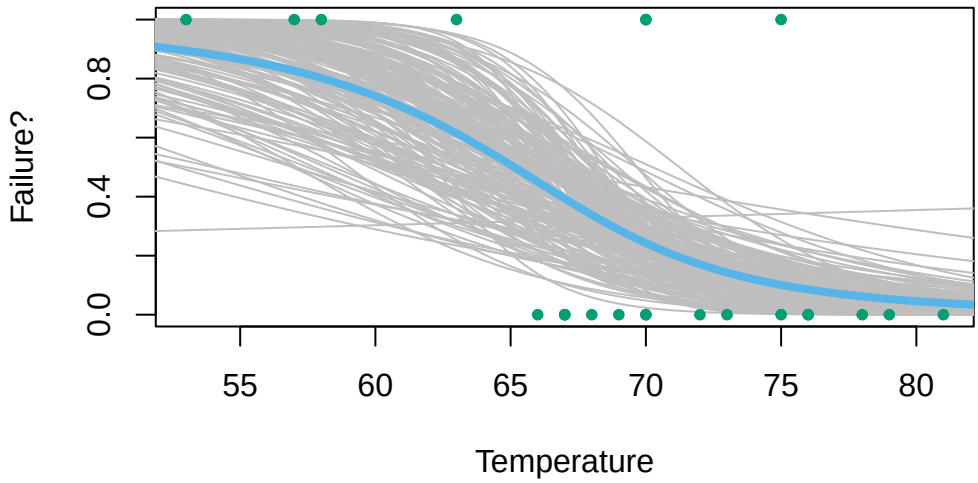

Figure 1: Posterior samples of the probability of failure.

Samples are given in Figure 1. We see that our Bayesian robot believes that it is extremely unlikely that lower temperatures are associated with a higher chance of failure and, indeed, that in most cases the failure of the O-rings is basically a foregone conclusion. On the day of the launch, the temperature was forecast to be 30 degrees, well below any of the experimental data. While we should always be wary of extrapolating beyond the range of our data, our robot would have made the following prediction for the probability of failure.

```
predict(challenger_bayes,
        newdata = data.frame(Temperature = 30),
        type = 'response')
```

```
        1
0.9838199
```

That is, the robot believes that the shuttle will experience an O-ring failure with probability roughly 98%.

5 Poisson Log-Linear Models

Another particular case of the generalized linear model takes

$$Y_i \sim \text{Poisson}(\mu_i) \qquad \text{where} \qquad \log(\mu_i) = x_i^\top \beta.$$

This is referred to as a *Poisson log-linear model*. Equivalently, we have $\mu_i = \exp(x_i^\top \beta)$.

Recall that a Poisson distribution is often used to model the *number of times an event occurs in a given time, or within a given space.* For example, it might be used to model the number of homicides in a city over a year, the number of goals scored in a soccer game, etc. This is used to model outcomes that are *counts* taking values in $0, 1, 2, \ldots$ with no obvious upper bound.

What the Coefficients Represent

Exercise 10: Coefficients in a Poisson Regression

Suppose we fit a Poisson log-linear model $\log(\mu_i) = \beta_0 + \beta_{i1}X_{i1} + \beta_{i2}X_{i2}$. Show that a change in X_{i2} by δ units, holding X_{i1} fixed, results in a *multiplicative effect on the mean*:

$$\mu_{\text{new}} = e^{\beta_2 \delta}\mu_{\text{old}}$$

Poisson Log-Linear Regression: The Ships Dataset

This example is taken from Section 6.3.2 of McCullaugh and Nelder (1989). We consider modeling the rate of reported damage incidents of certain types of cargo-carrying ships. The data is available in the `MASS` package and can be loaded as follows.

```
ships <- MASS::ships
head(ships)
```

```
  type year period service incidents
1    A   60     60     127         0
2    A   60     75      63         0
3    A   65     60    1095         3
4    A   65     75    1095         4
5    A   70     60    1512         6
6    A   70     75    3353        18
```

The variable `type` refers to the type of vessel, `year` to year in which the vessel was constructed, `period` to the period of time under consideration, and `service` to the number of months of service of all vessels of this type. The response of interest, `incidents`, refers to the total number of damage incidents which occurred during the period across *all* vessels constructed in year `year` and of type `type`; the reason for this pooling is that it is assumed that incidents occur according to a *Poisson process* with no ship-specific effects (possibly a dubious assumption, but it is all we can do with the data we have been given).

We are interested in three questions:

1. Do certain types of ships tends to have higher numbers of incidents, after controlling for other factors?
2. Were some periods more prone to other incidents, after controlling for other factors?
3. Did ships built in certain years have more accidents than others?

One possible choice of model we could use is a Poisson log-linear model of the form $\texttt{incidents}_i \sim \text{Poisson}(\mu_i)$ with

$$\log \mu_i = \beta_0 + \beta_{\texttt{service}} \cdot \texttt{service}_i + \beta_{\texttt{type}} \cdot \texttt{type}_i + \beta_{\texttt{period}} \cdot \texttt{period}_i + \beta_{\texttt{year}} \cdot \texttt{year}_i.$$

This model is fine, but we actually have more information about how to incorporate `service`: consider two ships, one of which was at service for 6 months and the other for a year, but which are otherwise identical. If the incidents really follow a homogeneous Poisson process, we would expect that the second ship has *twice as many* incidents as the first, on average. If this is the case, we should prefer the model

$$\log \mu_i = \beta_0 + \log(\texttt{service}_i) + \beta_{\texttt{type}} \cdot \texttt{type}_i + \beta_{\texttt{period}} \cdot \texttt{period}_i + \beta_{\texttt{year}} \cdot \texttt{year}_i.$$

Equivalently, we have $\mu_i = \texttt{service}_i \cdot \eta_i$ where η_i does not depend on $\texttt{service}_i$, giving the desired effect: doubling $\texttt{service}_i$ will double the mean. The term $\log(\texttt{service}_i)$ is called an *offset*; terms of this nature are very common in Poisson GLMs.

We can fit this model by maximum likelihood as follows.

```r
ships_glm <- glm(
  incidents ~ type + factor(period) + factor(year),
  family = poisson,
  offset = log(service),
  data = dplyr::filter(ships, service > 0)
)

print(summary(ships_glm))
```

```
Call:
glm(formula = incidents ~ type + factor(period) + factor(year),
    family = poisson, data = dplyr::filter(ships, service > 0),
    offset = log(service))

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)      -6.40590    0.21744 -29.460  < 2e-16 ***
typeB            -0.54334    0.17759  -3.060  0.00222 **
```

```
typeC            -0.68740    0.32904  -2.089  0.03670 *
typeD            -0.07596    0.29058  -0.261  0.79377
typeE             0.32558    0.23588   1.380  0.16750
factor(period)75  0.38447    0.11827   3.251  0.00115 **
factor(year)65    0.69714    0.14964   4.659 3.18e-06 ***
factor(year)70    0.81843    0.16977   4.821 1.43e-06 ***
factor(year)75    0.45343    0.23317   1.945  0.05182 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 146.328  on 33  degrees of freedom
Residual deviance:  38.695  on 25  degrees of freedom
AIC: 154.56

Number of Fisher Scoring iterations: 5
```

Notice that, like with a linear model or an ANOVA model, when controlling for categorical predictors we set one of the levels of the predictor to be a "reference" level. That is why we get coefficients for `typeB` through `typeE`, but not `typeA` as the `A` type is assumed to have a coefficient equal to zero.

From this we see that there is substantial evidence for the relevance of all variables. There is quite strong evidence for an effect of period, with a period of 75 being associated with more incidents. Similarly, it seems that incidents are particularly low for ships operating in year 60 relative to other years. Finally, there is evidence for differences across types of ships, with (for example) B having fewer incidents than A.

Exercise 11: Bayesian Poisson Loglinear Model

Fit this function using `stan_glm`, then try out the `plot` function for `stanreg` objects. Describe your results.

Exercise 12: Overdispersion

A problem with Poisson log-linear models is that they impose the restriction $\mathbb{E}(Y_i) = \text{Var}(Y_i)$ so that the variance is completely constrained by the mean. Count data is referred to as *overdispersed* if $\text{Var}(Y_i) > \mathbb{E}(Y_i)$.

a. Consider the model $Y \sim \text{Poisson}(\lambda)$ (given λ) and $\lambda \sim \text{Gam}(k, k/\mu)$. Find the mean and variance of Y. Is Y overdispersed?

b. Show that Y marginally has a negative binomial distribution with k failures and

success probability $\mu/(k+\mu)$; recall that the negative binomial distribution has mass function

$$f(y \mid k, p) = \binom{k+y-1}{y} p^y (1-p)^k.$$

c. The following data is taken from Table 14.6 in Categorical Data Analysis, 3rd edition, by Alan Agresti.

```
poisson_data <- data.frame(
  Response = 0:6,
  Black = c(119,16,12,7,3,2,0),
  White = c(1070,60,14,4,0,0,1)
)
knitr::kable(poisson_data, booktabs = TRUE)
```

Response	Black	White
0	119	1070
1	16	60
2	12	14
3	7	4
4	3	0
5	2	0
6	0	1

The data is from a survey of 1308 people in which they were asked how many homicide victims they know. The variables are `response`, the number of victims the respondent knows, and `race`, the race of the respondent (black or white). The question is: to what extend does race predict how many homicide victims a person knows?

For this data, is it true that the mean outcome (for either black or white individuals) is approximately equal to its variance? If not, do we see overdispersion or underdispersion?

d. Using the fact that the variance of Y_i under a negative binomial is $\mu+\mu^2/k$, compute an estimate of k for Black and White individuals for the two groups. Does the same value of k seem appropriate for both groups, or does on group seem to have a larger value of k than the other? (Don't worry about quantifying uncertainty in this assessment.)