C3M2: Peer Reviewed Assignment

Outline:

The objectives for this assignment:

1. Apply Poisson Regression to real data.
2. Learn and practice working with and interpreting Poisson Regression Models.
3. Understand deviance and how to conduct hypothesis tests with Poisson Regression.
4. Recognize when a model shows signs of overdispersion.

General tips:

1. Read the questions carefully to understand what is being asked.
2. This work will be reviewed by another human, so make sure that you are clear and concise in what your explanations and answers.

In [1]:
```
# Load the required packages
library(MASS)
```

Problem 1: Poisson Estimators

Let $Y_1, \ldots, Y_n \overset{i}{\sim} Poisson(\lambda_i)$. Show that, if $\eta_i = \beta_0$, then the maximum likelihood estimator of λ_i is $\widehat{\lambda}_i = \bar{Y}$, for all $i = 1, \ldots, n$.

To solve this problem, we first need to understand the relationship between the variables and the likelihood function.

Given $Y_1, \ldots, Y_n \overset{i}{\sim} \text{Poisson}(\lambda_i)$, the probability mass function of each Y_i is given by:

$$P(Y_i = y_i) = \frac{\lambda_i^{y_i} e^{-\lambda_i}}{y_i!}$$

The joint likelihood function for all Y_is is:

$$L(\lambda_1, \ldots, \lambda_n \mid Y_1, \ldots, Y_n) = \prod_{i=1}^{n} \frac{\lambda_i^{y_i} e^{-\lambda_i}}{y_i!}$$

Taking the logarithm of the likelihood function (log-likelihood function):

$$\ell(\lambda_1, \ldots, \lambda_n \mid Y_1, \ldots, Y_n) = \sum_{i=1}^{n} \left(y_i \log(\lambda_i) - \lambda_i - \log(y_i!) \right)$$

We are asked to show that the maximum likelihood estimator (MLE) of λ_i is $\widehat{\lambda}_i = \bar{Y}$ for all $i = 1, \ldots, n$ under the condition that $\eta_i = \beta_0$.

Step 1: Define the Relationship Between λ_i and η_i

If $\eta_i = \beta_0$, and $\lambda_i = e^{\eta_i}$, then:

$$\lambda_i = e^{\beta_0}$$

This implies that all λ_is are equal, i.e., $\lambda_1 = \lambda_2 = \cdots = \lambda_n = \lambda$, where $\lambda = e^{\beta_0}$.

Step 2: Simplify the Log-Likelihood Function

Under this condition, the log-likelihood function simplifies to:

$$\ell(\lambda \mid Y_1, \ldots, Y_n) = \sum_{i=1}^{n} \left(y_i \log(\lambda) - \lambda - \log(y_i!) \right)$$

This can be written as:

$$\ell(\lambda \mid Y_1, \ldots, Y_n) = \left(\sum_{i=1}^{n} y_i \right) \log(\lambda) - n\lambda - \sum_{i=1}^{n} \log(y_i!)$$

Step 3: Find the MLE

To find the MLE of λ, we take the derivative of the log-likelihood function with respect to λ and set it equal to zero:

$$\frac{\partial \ell(\lambda \mid Y_1, \ldots, Y_n)}{\partial \lambda} = \frac{\sum_{i=1}^{n} y_i}{\lambda} - n = 0$$

Solving for λ:

$$\lambda = \frac{1}{n} \sum_{i=1}^{n} y_i = \bar{Y}$$

Conclusion:

Since $\lambda = e^{\beta_0}$, and the MLE for λ under the given condition is $\bar{Y}$, we have $\widehat{\lambda}_i = \bar{Y}$ for all $i = 1, \ldots, n$.

Thus, the maximum likelihood estimator of λ_i is $\widehat{\lambda}_i = \bar{Y}$ for all $i = 1, \ldots, n$ when $\eta_i = \beta_0$.

Problem 2: Ships data

The ships dataset gives the number of damage incidents and aggregate months of service for different types of ships broken down by year of construction and period of operation.

The code below splits the data into a training set (80% of the data) and a test set (the remaining 20%).

In [3]:
```
data(ships)
ships = ships[ships$service != 0,]
ships$year = as.factor(ships$year)
ships$period = as.factor(ships$period)

set.seed(11)
n = floor(0.8 * nrow(ships))
index = sample(seq_len(nrow(ships)), size = n)

train = ships[index, ]
test = ships[-index, ]
head(train)
summary(train)
```

A data.frame: 6 × 5

	type	year	period	service	incidents
	<fct>	<fct>	<fct>	<int>	<int>
40	E	75	75	542	1
28	D	65	75	192	0
18	C	60	75	552	1
19	C	65	60	781	0
5	A	70	60	1512	6
32	D	75	75	2051	4

```
 type   year    period      service          incidents     
 A:5    60:7    60:11    Min.    :   45.0   Min.    : 0.00  
 B:5    65:8    75:16    1st Qu.:  318.5   1st Qu.: 0.50  
 C:6    70:8             Median : 1095.0   Median : 2.00  
 D:7    75:4             Mean   : 5012.2   Mean   :10.63  
 E:4                     3rd Qu.: 2202.5   3rd Qu.:11.50  
                         Max.   :44882.0   Max.   :58.00  
```

2. (a) Poisson Regression Fitting

Use the training set to develop an appropriate regression model for `incidents`, using `type`, `period`, and `year` as predictors (HINT: is this a count model or a rate model?).

Calculate the mean squared prediction error (MSPE) for the test set. Display your results.

In [5]:
```
# Your Code Here# Fit the Poisson regression model
model <- glm(incidents ~ type + period + year,
             data = train,
             family = poisson)

summary(model)


# Predict incidents on the test set
predicted_incidents <- predict(model, newdata = test, type = "response")

# Calculate the mean squared prediction error (MSPE)
mspe <- mean((test$incidents - predicted_incidents)^2)
mspe
```

```
Call:
glm(formula = incidents ~ type + period + year, family = poisson,
    data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-4.0775  -1.9869  -0.0418   0.7612   3.6618

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   1.5644     0.2199   7.113 1.13e-12 ***
typeB         1.6795     0.1889   8.889  < 2e-16 ***
typeC        -2.0789     0.4408  -4.717 2.40e-06 ***
typeD        -1.1551     0.2930  -3.943 8.06e-05 ***
typeE        -0.5113     0.2781  -1.839   0.0660 .
period75      0.4123     0.1282   3.216   0.0013 **
year65        0.4379     0.1885   2.324   0.0201 *
year70        0.2260     0.1916   1.180   0.2382
year75        0.1436     0.3147   0.456   0.6481
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 554.70  on 26  degrees of freedom
Residual deviance: 109.21  on 18  degrees of freedom
AIC: 200.92

Number of Fisher Scoring iterations: 6
```

131.077556337426

The mean squared prediction error (MSPE) for the test set, using the Poisson regression model, is 131.08. The residual deviance is 109.21

2. (b) Poisson Regression Model Selection

Do we really need all of these predictors? Construct a new regression model leaving out `year` and calculate the MSPE for this second model.

Decide which model is better. Explain why you chose the model that you did.

In [6]:
```
# Your Code Here
# Fit the Poisson regression model without 'year'
model_no_year <- glm(incidents ~ type + period,
                     data = train,
                     family = poisson)

# Predict incidents on the test set
predicted_incidents_no_year <- predict(model_no_year, newdata = test, type = "res
ponse")

# Calculate the mean squared prediction error (MSPE)
mspe_no_year <- mean((test$incidents - predicted_incidents_no_year)^2)
mspe_no_year
```

275.122550627591

In [8]:
```
# Can compare nested poisson models with a chi-squared
# Fit the Poisson regression model without 'year'
model_no_year <- glm(incidents ~ type + period,
                     data = train,
                     family = poisson)

# Calculate the chi-squared test statistic for model comparison
p_value <- pchisq(model_no_year$deviance - model$deviance,
                  df = model_no_year$df.residual - model$df.residual,
                  lower.tail = FALSE)

p_value
```

0.0929203838345225

Since the p-value is greater than the conventional threshold of 0.05, we failed to reject the null hypothesis ,suggesting that removing the year variable does not result in a significantly worse model fit.

But the mspe shows that the full model with year behaves better

2. (c) Deviance

How do we determine if our model is explaining anything? With linear regression, we had a F-test, but we can't do that for Poisson Regression. If we want to check if our model is better than the null model, then we're going to have to check directly. In particular, we need to compare the deviances of the models to see if they're significantly different.

Conduct two χ^2 tests (using the deviance). Let $\alpha = 0.05$:

1. Test the adequacy of null model.
2. Test the adequacy of your chosen model against the full model (the model fit to all predictors).

What conclusions should you draw from these tests?

In [12]:
```
# Your Code Here
# Test if the model is better than the null model
model_null <- glm(incidents ~ 1,
                  data = train,
                  family = poisson)
# Test chi_sq stat
# Chi-squared test comparing the null model to the chosen model
p_value_null <- pchisq(model_null$deviance - model$deviance,
                       df = model_null$df.residual - model$df.residual,
                       lower.tail = FALSE)

p_value_null

chisq.stat = with(train, sum((incidents - fitted(model))^2/fitted(model)))
pchisq(chisq.stat, df=model$df.residual, lower.tail=FALSE)

# Test against the full model
# Fit the saturated model (model with maximum possible predictors)
model_saturated <- glm(incidents ~ .,
                       data = train,
                       family = poisson)

# Chi-squared test comparing the chosen model to the saturated model
p_value_saturated <- pchisq(model$deviance - model_saturated$deviance,
                            df = model$df.residual - model_saturated$df.residual,
                            lower.tail = FALSE)

p_value_saturated
```

3.41896472956775e-91

4.22139949448423e-13

1.85320875968548e-19

The p-value for the test comparing the null model to the chosen model is 3.42e-91, indicating that the chosen model is **significantly better than the null model**. This suggests that the predictors in the chosen model do explain a substantial amount of variation in the number of incidents.

The p-value for the test comparing the chosen model to the saturated model is 1.85e-19, indicating that while the chosen model is **not as good as the saturated model**, it still provides a statistically significant fit to the data. This confirms that the chosen model is effective but not perfect.

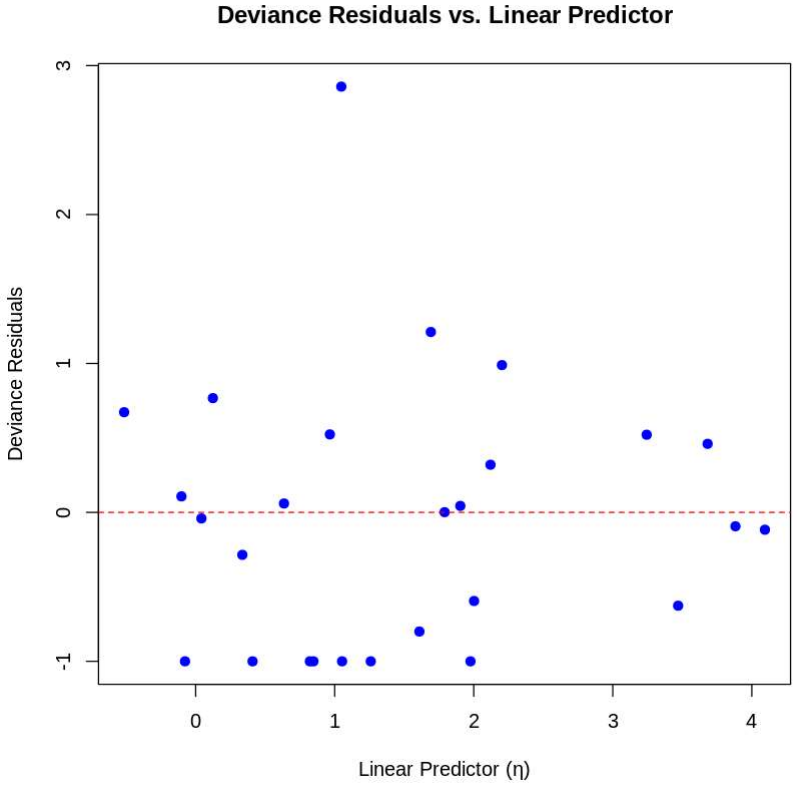
2. (d) Poisson Regression Visualizations

Just like with linear regression, we can use visualizations to assess the fit and appropriateness of our model. Is it maintaining the assumptions that it should be? Is there a discernable structure that isn't being accounted for? And, again like linear regression, it can be up to the user's interpretation what is an isn't a good model.

Plot the deviance residuals against the linear predictor η. Interpret this plot.

In [14]:
```r
# Your Code Here
# Plot the deviance residuals against the linear predictor
plot(x = model$linear.predictors,
     y = model$residuals,
     xlab = "Linear Predictor (η)",
     ylab = "Deviance Residuals",
     main = "Deviance Residuals vs. Linear Predictor",
     pch = 19,
     col = "blue")

# Add a horizontal line at y=0 for reference
abline(h = 0, lty = 2, col = "red")
```



Overall, the plot suggests that the model is generally appropriate for the data, but the outlier with a high positive residual could indicate that the model might benefit from further refinement or that there is an exceptional case in the data.

In this updated plot, there is a clear outlier with a deviance residual around 3 at a linear predictor η value close to 0. This suggests that the model significantly underpredicts the number of incidents for this observation.

2. (e) Overdispersion

For linear regression, the variance of the data is controlled through the standard deviation σ, which is independent of the other parameters like the mean μ. However, some GLMs do not have this independence, which can lead to a problem called overdispersion. Overdispersion occurs when the observed data's variance is higher than expected, if the model is correct.

For Poisson Regression, we expect that the mean of the data should equal the variance. If overdispersion is present, then the assumptions of the model are not being met and we can not trust its output (or our beloved p-values)!

Explore the two models fit in the beginning of this question for evidence of overdisperion. If you find evidence of overdispersion, you do not need to fix it (but it would be useful for you to know how to). Describe your process and conclusions.

In [15]:
```
# Your Code Here
# 1. Dispersion statistic for the full model (with 'year')
dispersion_full_model <- model$deviance / model$df.residual
dispersion_full_model

# 2. Dispersion statistic for the model without 'year'
dispersion_no_year_model <- model_no_year$deviance / model_no_year$df.residual
dispersion_no_year_model
```

6.06734885050602

5.50624192534451

Interpretation of the Dispersion Statistics

The calculated dispersion statistics for the two models are as follows:

1. **Full Model (with `year`)**: 6.07
2. **Model without `year`**: 5.51

Conclusion

Both models exhibit substantial overdispersion, as indicated by the dispersion statistics being significantly greater than 1. Specifically:

- **Full Model (with `year`)**: A dispersion statistic of 6.07 means that the variance of the data is more than six times greater than what the Poisson model assumes. This strongly indicates overdispersion.
- **Model without `year`**: A dispersion statistic of 5.51 suggests a similar level of overdispersion, indicating that the problem persists even after removing the `year` variable.

Implications

- **Model Assumptions Violated**: The assumption that the mean equals the variance in the Poisson model is not holding. This overdispersion suggests that the model may not be fully appropriate for the data, and as a result, the p-values and confidence intervals derived from these models may be unreliable.
- **Next Steps**: To address overdispersion, you might consider alternative models such as:
  - **Quasi-Poisson Regression**: Adjusts the standard errors to account for overdispersion while retaining the Poisson structure.
  - **Negative Binomial Regression**: Explicitly models overdispersion by adding an additional parameter to account for the excess variance.

Understanding and addressing overdispersion is crucial to ensuring that the inferences drawn from the model are valid.