# 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 [3]:

```
# 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 $\lambda_i$ is $\widehat{\lambda}_i = \bar{Y}$, for all $i = 1, \ldots, n$.

First, note that if $\eta_i = \beta_0$ for all $i = 1, \ldots, n$, then we can drop the $i$ subscript. That is, we are not estimating a different mean for each response because there are no predictors. From class, we know that the log-likelihood for Poisson regression is:

$$\ell(\beta_0) = \sum_{i=1}^{n} \left[ y_i \eta - e^{\eta} - \log(y_i!) \right] = \sum_{i=1}^{n} \left[ y_i \beta_0 - e^{\beta_0} - \log(y_i!) \right] =$$

$$\sum_{i=1}^{n} \left[ y_i \beta_0 - e^{\beta_0} - \log(y_i!) \right]$$

To find the MLE, we maximize the log-likelihood with respect to $\beta_0$; the maximizer is the MLE. To find the MLE of $\lambda$, we use the relationship that $\lambda = e^{\beta_0}$.

We take the derivative of $\ell$, set it equal to zero, and solve for :

$$\frac{d\ell(\beta_0)}{d\beta_0} = \sum_{i=1}^{n} \left[ y_i \beta_0 - e^{\beta_0} - \log(y_i!) \right] = \sum_{i=1}^{n} y_i - \sum_{i=1}^{n} e^{\widehat{\beta}_0} \overset{set}{=} 0$$

$$\implies \sum_{i=1}^{n} y_i = \sum_{i=1}^{n} e^{\widehat{\beta}_0} \implies \sum_{i=1}^{n} y_i = n e^{\widehat{\beta}_0}$$

$$\implies \frac{1}{n} \sum_{i=1}^{n} y_i = e^{\widehat{\beta}_0} \implies \bar{y} = e^{\widehat{\beta}_0}$$

$$\implies \widehat{\beta}_0 = \log(\bar{y}).$$

So, the MLE (as a random variable) of $\beta_0$ is $\widehat{\beta}_0 = \log(\bar{Y})$. By the invariance property of MLEs, the MLE of $\lambda = e^{\beta_0}$ is

$$\widehat{\lambda} = e^{\widehat{\beta}_0} = e^{\log(\bar{Y})} = \bar{Y}.$$

# Problem 2:

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 [4]:

```
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)
```

| | type | year | period | service | incidents |
|---|---|---|---|---|---|
| **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 [12]:
```
# Your Code Here
model = glm(incidents~type+period+year, train, family="poisson")
summary(model)

p = predict(model, test, type = "response")
MSPE = mean((test$incidents - p)^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
```

```
131.077556337427
```

We fit the model using poisson regression. This resulted in a model with residual deviance of 109.21 and an MSPE of $\approx 131$.

## 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 MSE for this second model.

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

In [14]:

```
# Your Code Here
model.small = glm(incidents~type+period, train, family="poisson")
summary(model.small)
p.small = predict(model.small, test, type = "response")
MSPE.small = mean((test$incidents - p.small)^2)
MSPE.small
```

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

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-4.2377  -1.9003  -0.1372   0.6377   3.8906  

Coefficients:
            Estimate Std. Error z value Pr(>|z|)    
(Intercept)   1.7190     0.1838   9.355  < 2e-16 ***
typeB         1.7831     0.1781  10.014  < 2e-16 ***
typeC        -2.0573     0.4394  -4.683 2.83e-06 ***
typeD        -1.1281     0.2918  -3.866 0.000111 ***
typeE        -0.4831     0.2767  -1.746 0.080787 .  
period75      0.4723     0.1222   3.865 0.000111 ***
---
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: 115.63  on 21  degrees of freedom
AIC: 201.34

Number of Fisher Scoring iterations: 6
```

275.122550627591

In [7]:

```
# Can compare nested poisson models with a chi-squared
pchisq(model.small$deviance-model$deviance, df=model.small$df.resid
ual-model$df.residual, lower.tail=FALSE)
```

0.0929203838345219

The chi-squared test has a p-value of $0.09 > \alpha$, so we fail to reject the null at $\alpha = 0.05$ and might conclude that the reduced model is sufficient. However, the MSPE of the model without `year` is much larger than the model with `year` as a predictor. So, if the goal of our model is prediction, then the full model appears 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 $\chi^2$ tests (using the deviance). Let $\alpha = 0.05$:

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

What conclusions should you draw from these tests?

In [18]:

```r
# Your Code Here
# Test if the model is better than the null model
chisq.stat = with(train, sum((incidents - fitted(model))^2/fitted(model)))
# Test chi_sq stat
pchisq(chisq.stat, df=model$df.residual, lower.tail=FALSE)

# Test against the saturated model
model.sat = glm(incidents~., train, family="poisson")
pchisq(model$deviance-model.sat$deviance, df=model$df.residual-model.sat$df.residual, lower.tail=FALSE)
```

```
4.22139949448438e-13

1.85320875968549e-19
```

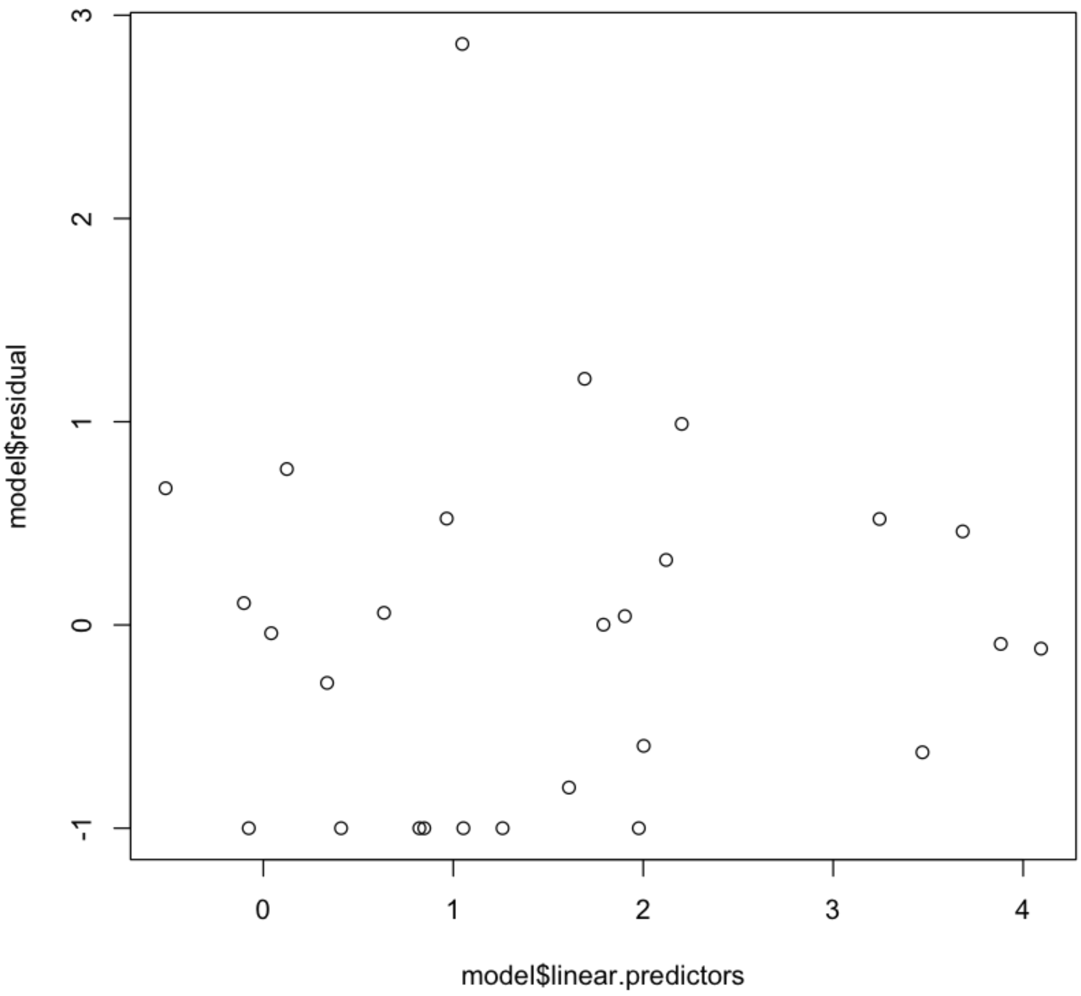
We get a statistically signficant result from both tests, meaning our model is better than the null model but worse than the saturated model.

## 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 $\eta$. Interpret this plot.

```
In [21]: # Your Code Here
         plot(x=model$linear.predictors, y=model$residual)
```



The residual vs fitted plot looks ok. There appears to be an outlier residual just below $3$.

## 2. (e) Overdispersion

For linear regression, the variance of the data is controlled through the standard deviation $\sigma$, which is independent of the other parameters like the mean $\mu$. 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 [22]:

```
# Your Code Here
summary(model)
summary(model.small)
```

```
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
```

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

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-4.2377  -1.9003  -0.1372   0.6377   3.8906

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   1.7190     0.1838   9.355  < 2e-16 ***
typeB         1.7831     0.1781  10.014  < 2e-16 ***
typeC        -2.0573     0.4394  -4.683 2.83e-06 ***
typeD        -1.1281     0.2918  -3.866 0.000111 ***
typeE        -0.4831     0.2767  -1.746 0.080787 .
period75      0.4723     0.1222   3.865 0.000111 ***
---
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: 115.63  on 21  degrees of freedom
AIC: 201.34

Number of Fisher Scoring iterations: 6
```

The rule of thumb is that overdispersion is present if the Residual deviance is significantly greater than the degrees of freedom. Because this is true for both models, both would be considered to have overdispersion.