Poisson Regression

In this learning module, you'll be familiarized with **Poisson Regression** as a method for modeling *count data*. You'll frequently encounter a count outcomes, for example:

- The number of tickets sold for a concert
- The number of elected Republican congress members
- The number of crimes that occur in a particular area

As you can imagine, you would want to leverage a statistical technique that assumes a **distribution of counts** for the outcome variable. One distrubtion that meets this constraint is the **poisson distribution**, which can be the assumed outcome distrubution of a **generalized linear model**. In the sections below, you'll review the poisson distribution, implement a Poisson regression, and interpret the results.

Resoures

You may find the following resources helpful in learning about Poisson regression:

- UCLA Poisson Example
- Wikipedia: Poisson Regression
- Regression Models for Data Science: Poisson Regression

Poisson Distribution

As you may recall from earlier in the course, the **Poisson Distribution** is a distribution of count values. The distribution is described by a **single parameter** lambda (λ). Note,

> In a Poisson distribution, the **mean** is equal to the **variance**. This is captured in the single parameter, lambda (λ).

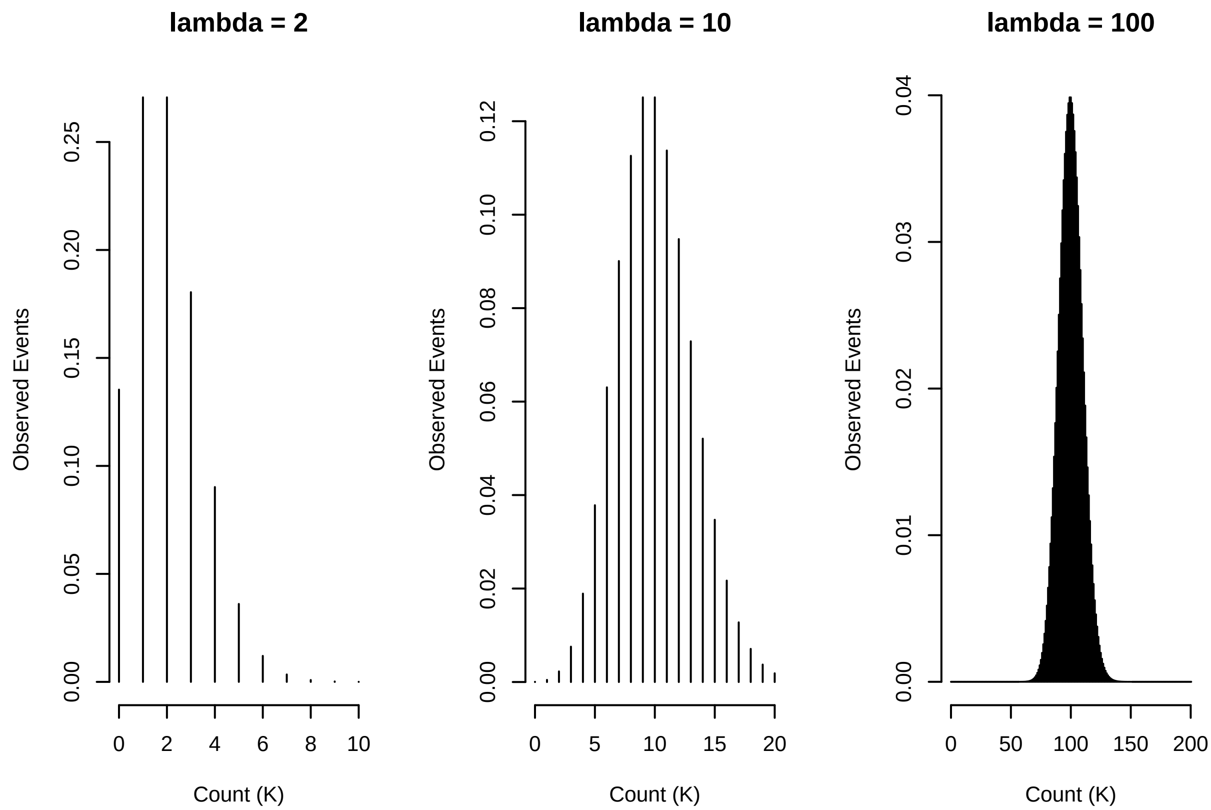
The expected probability of observing K events is defined as:

$$P\left(k\ events\right) = \frac{\lambda^k e^{-\lambda}}{k!}$$

Given the above formula, the probaility of observing 2 events if the mean number of events is 3 ($\lambda = 3$) is:

$$P\left(3\ events\right) = \frac{3^2 e^{-3}}{2!} = .22$$

Below are the randomly drawn Poisson distributions for $\lambda = 2$, $\lambda = 10$, $\lambda = 100$ (code found here:



When modeling count data, it is important to **check the distribution** of your outcome variable to see how well it follows a Poisson distribution. If it does not, you may want to implement a Negative Binomial Regression (example), which loosens the assumption that the mean and variance are equal. You can also consider a Zero-Inflated Poisson Regression (example), or Zero-Inflated Negative Binomial Regression (example).

Poisson Formula

The formula for fitting an outcome variable with a Poisson distribution is a type of **Generalized Linear Model**. The **link** between the **linear** set of inputs and the output is a **log-link**. At a glance, this seems similar to logging the outcome variable. However, directly modeling the log of the outcome is not possible if(when) the *count is zero* ($log(0) = -Inf$). As such, Poisson Regressions model the **log of the expected value** of the outcome, given a vector of input variables (source:

$$log\,(E[Y_i|X_i = x_i]) = log\,(\mu_i) = B_0 + B_1 x_i$$

In the equation above, the **log of the expected value of Y_i given X_i** is linearly approximated using $B_0 + B_1 x_i$.

Similarly to Logistic regression, the coefficients (betas) are obtained through a *Maximum Likelihood Estimation* that seeks to produce a formula that maximizes the probability of observing the data. While this MLE procedure is beyond the scope of this course, Poisson models are easily implemented in R or Python.

Generalized Linear Models

As noted above, Poisson models are in the family of Generalized Linear Models. The following excerpt from this book:

" *Generalized linear modeling* is a framework for statistical analysis that includes linear and logistic regression as special cases. Linear regression directly predictes continuous data y from a *linear predictor* $X\beta = \beta_0 + X_1\beta_1 + ... + +X_k\beta_k$. Logistic regression predicts $Pr(y = 1)$ for binary data from a linear preditor with an inverse-logit transofmration. A generalized linear model involves:

1. A data vecotr $y = (y_1, ..., y_n)$
2. Precitors X cand coefficnets β, forming a inear predictor $X\beta$
3. A *link function* g, yielding a vector of transofrmed data $\hat{y} = g^{-1}(X\beta)$ that are used to model the data
4. A data stirubtion, $p(y|\hat{y})$
5. Possibly other parameters, such as variances, overdispersions, and cutpoints, involved in the predictors, link function, and data distribution." (p.109)

As such, we assume a **Poisson distribution** and use a *logartithmic transformation* as the link for a Poisson regression, which allows the set of predicted values ($\hat{y}$)to be **positive**.

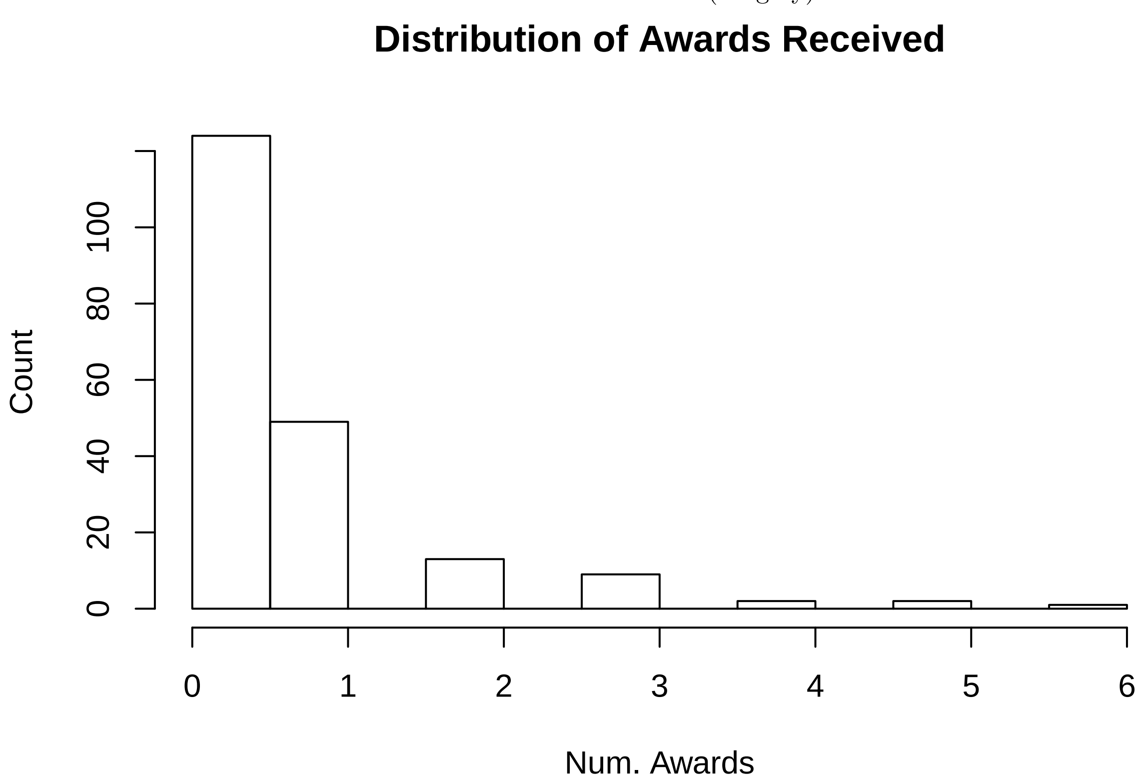
Implementation

Implementing a Poisson model in R or Python is a straightforward procedure. Using the generalized linear model function (`glm`), a model can be easily implemented:

```
m1 <- glm(outome ~ var1 + var2, family="poisson", data=df)
```

Here, the model (`m1`) is estimating coefficients for independent variables (`var1` and `var2`) for predicting the *mean expected value* of an outcome variable (`outcome`). Below is an example from this website of predicting **number of student awards** based on *type of program* (`prog`) the student is enrolled in (vocational, general or academic) and their score on a final exam in math (`math`).

We can see that the distribution of the number of awards is (roughly) Poisson:



It is then straightforward to create a Poisson model:

```r
m1 <- glm(num_awards ~ prog + math, family="poisson", data=p)
print(summary(m1))
```

```
##
## Call:
## glm(formula = num_awards ~ prog + math, family = "poisson", data = p)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2043  -0.8436  -0.5106   0.2558   2.6796
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -5.24712    0.65845  -7.969 1.60e-15 ***
## progAcademic    1.08386    0.35825   3.025  0.00248 **
## progVocational  0.36981    0.44107   0.838  0.40179
## math            0.07015    0.01060   6.619 3.63e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 287.67  on 199  degrees of freedom
## Residual deviance: 189.45  on 196  degrees of freedom
## AIC: 373.5
##
## Number of Fisher Scoring iterations: 6
```