Chapter 1

2025-07-03

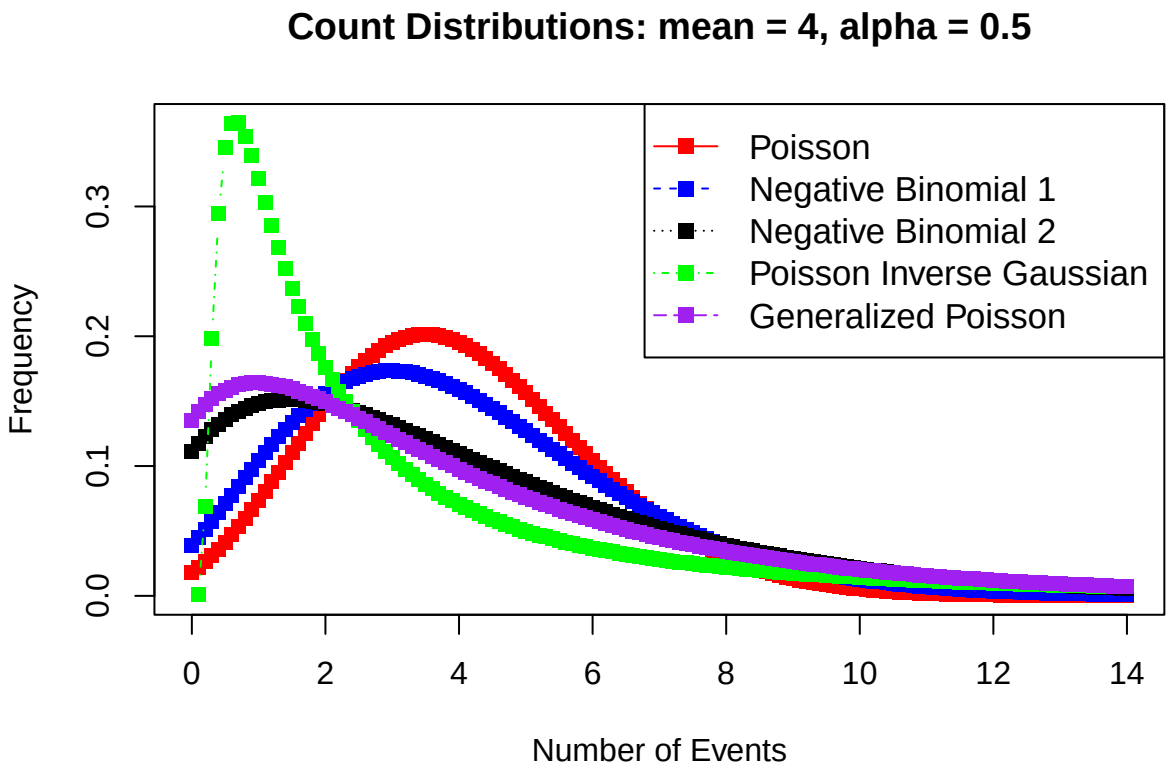
```
## Code for commonly used count distributions
obs <- 15;
mu <- 4;
y <- (0:140)/10;
alpha <- .5;
amu <- mu*alpha; layout(1)
all.lines <- vector(mode = 'list', length = 5)
for (i in 1:length(mu)) {
  # Poisson
  yp = exp(-mu[i])*(mu[i]^y) / factorial(y)
  # Neg Binomial 1
  ynb1 = exp(log(gamma(mu[i]/alpha + y))
             - log(gamma(y+1))
             - log(gamma(mu[i]/alpha))
             + (mu[i]/alpha)*log(1/(1+alpha))
             +y*log(1-1/(1+alpha)))
  # Neg Binomial 2
  ynb2 = exp( y*log(amu[i]/(1+amu[i]))
              - (1/alpha) * log(1+amu[i])
              + log(gamma(y+1/alpha))
              - log(gamma(y+1))
              - log(gamma(1/alpha)) )
  # Poisson Inverse Gaussian
  ypig = exp( (-(y-mu[i])^2) / (alpha*2*y*mu[i]^2))*(sqrt(1/(alpha*2*pi*y^3)))
  # Generalized Poisson
  ygp = exp( log((1-alpha)*mu[i])
             + (y-1)*log((1-alpha) * mu[i]+alpha*y)
             - (1-alpha)*mu[i]
             - alpha * y
             - log(gamma(y+1)))

  all.lines = list(yp = yp, ynb1 = ynb1, ynb2 = ynb2, ypig = ypig, ygp = ygp)

  # Chart
  ymax = max(unlist(all.lines), na.rm=TRUE)
  cols = c("red", "blue", "black", "green", "purple")
  plot(y, all.lines[[1]], ylim = c(0, ymax), type="n", main="Count Distributions: mean = 4, alpha = 0.5
            xlab = "Number of Events",
            ylab = "Frequency",
      )
  for (j in 1:5)
  lines(y, all.lines[[j]], ylim = c(0, ymax), col = cols[j], type = "b", pch = 15, lty = j)

  legend('topright', cex = 1.1, pch = 15,
```

```
        legend = c("Poisson", "Negative Binomial 1", "Negative Binomial 2", "Poisson Inverse Gaussian"
        col = cols,
        lty = 1:5,
        lwd = 1)
}
```




```
## Fitting a Poisson Distribution
# Define the mean value
mu <- 2
# Define response
y <- c(4, 2, 0.3, 1, 2)
```

We generate Poisson probabilities using the Poisson probability distribution. The probability mass function of the Poisson distribution is:

$$f(y;\mu) = \frac{e^{-\mu_i}\mu_i^{y_i}}{y_i!}$$

- $y_i \in \{0, 1, 2, \dots\}$ is the observed count
- $\mu_i > 0$ is the expected rate (mean)
- $f(y;\mu)$ gives the probability of observing y

```
y0 <- exp(-mu) * (mu^0) / factorial(0)
y1 <- exp(-mu) * (mu^1) / factorial(1)
y2 <- exp(-mu) * (mu^2) / factorial(2)
y3 <- exp(-mu) * (mu^3) / factorial(3)
```

```r
y4 <- exp(-mu) * (mu^4) / factorial(4)
# Storing the Pois Prob
poisProb <- c(y0, y1, y2, y3, y4); poisProb
```

```
## [1] 0.13533528 0.27067057 0.27067057 0.18044704 0.09022352
```

Using inbuilt function in R: dpois for the PMF of a Poisson random variable

```r
dpois(0:4, lambda = mu)
```

```
## [1] 0.13533528 0.27067057 0.27067057 0.18044704 0.09022352
```

The function `ppois` in R returns the cumulative probability:

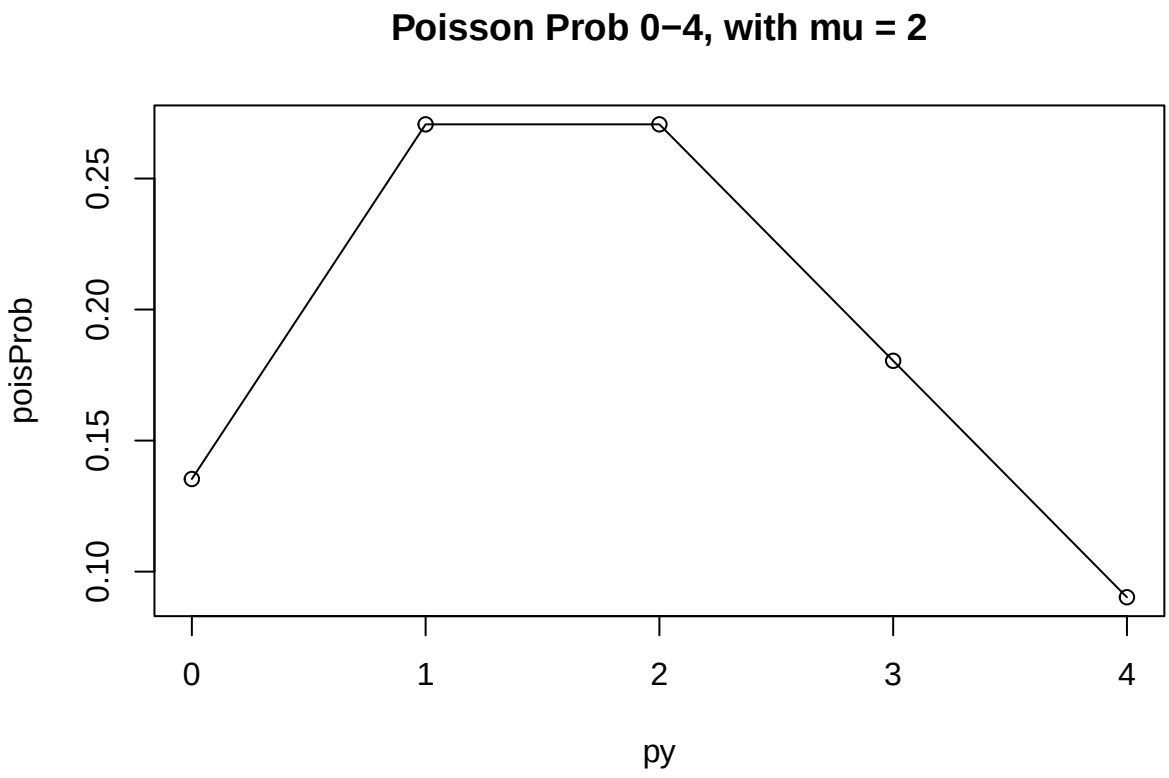
$$P(Y \leq q) = \sum_{k=0}^{q} \frac{e^{-\lambda}\lambda^k}{k!}$$

This is the probability that a Poisson random variable with mean λ takes a value **less than or equal to** q.

```r
ppois(0:4, lambda = mu)
```

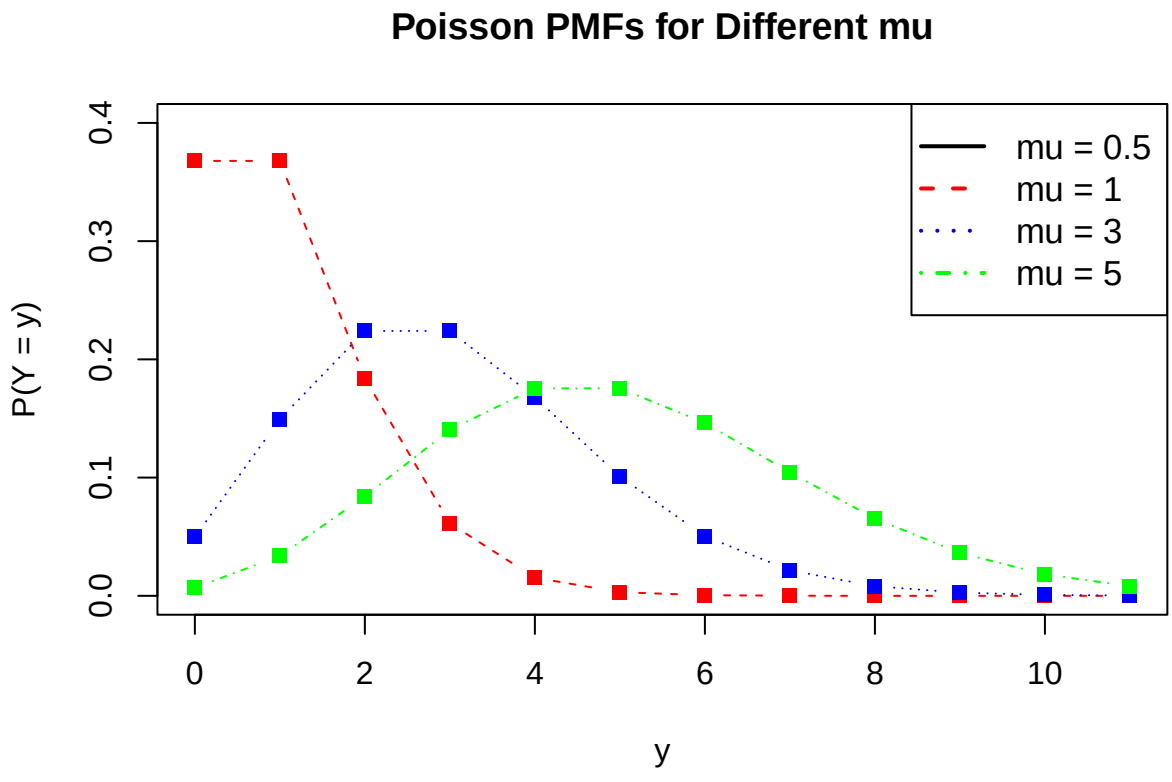
```
## [1] 0.1353353 0.4060058 0.6766764 0.8571235 0.9473470
```

```r
# Visualize Poisson probabilities
py <- 0:4
plot(poisProb ~ py, xlim=c(0,4), type="o", main ="Poisson Prob 0-4, with mu = 2")
```



```r
# Visualize for different mu
mus <- c(0.5, 1, 3, 5)
y <- 0:11
cols <- c("black", "red", "blue", "green")
layout(1)

for (i in 1:length(mus)) {
  p <- dpois(y, mus[i])
  if (i == 1) {
    plot(y, p, col = cols[i], type = 'n', lty = i,
         ylim = c(0, 0.4),
         pch = 15,
         main = "Poisson PMFs for Different mu",
         xlab = "y", ylab = "P(Y = y)")
  } else {
    lines(y, p, col = cols[i], lty = i, pch = 15, type = 'b')
  }
}
legend('topright', cex = 1.1, pch = NA,
       legend = c("mu = 0.5", "mu = 1", "mu = 3", "mu = 5"),
       col = cols,
       lty = 1:4,
       lwd = 2)
```



Maximum Likelihood Estimation for the Poisson Distribution

Let $y_1, y_2, \ldots, y_n$ be independent observations such that:

$$y_i \sim \text{Poisson}(\mu)$$

The probability mass function (PMF) is:

$$f(y_i; \mu) = \frac{e^{-\mu} \mu^{y_i}}{y_i!}$$

Log-Likelihood Function

The likelihood function is:

$$L(\mu) = \prod_{i=1}^{n} \frac{e^{-\mu} \mu^{y_i}}{y_i!}$$

Taking logs:

$$\ell(\mu) = \log L(\mu) = \sum_{i=1}^{n} \left[-\mu + y_i \log \mu - \log y_i! \right]$$

MLE for Constant Mean μ

Differentiate the log-likelihood:

$$\frac{d\ell}{d\mu} = -n + \frac{1}{\mu} \sum_{i=1}^{n} y_i$$

Set to zero and solve:

$$\hat{\mu}_{\text{MLE}} = \frac{1}{n} \sum_{i=1}^{n} y_i = \bar{y}$$

So the MLE of μ is simply the **sample mean**.

Poisson Regression with Log Link

Now suppose each observation has a vector of covariates $x_i \in \mathbb{R}^k$, and we model:

$$\mu_i = \exp(x_i^\top \beta), \quad \text{so that} \quad y_i \sim \text{Poisson}(\mu_i)$$

The log-likelihood becomes:

$$\ell(\beta) = \sum_{i=1}^{n} \left[-\exp(x_i^\top \beta) + y_i x_i^\top \beta - \log y_i! \right]$$

Score Function (Gradient, First Derivative)

Differentiating with respect to β:

$$\nabla \ell(\beta) = \sum_{i=1}^{n} (y_i - \mu_i) x_i = X^\top (y - \mu)$$

Where $X \in \mathbb{R}^{n \times k}$ is the design matrix and $\mu = \exp(X\beta)$ (elementwise exponential). Note it's also denoted with g.

Hessian (Second Derivative)

The Hessian is:

$$\nabla^2 \ell(\beta) = -X^\top W X$$

Where $W = \text{diag}(\mu_1, \ldots, \mu_n)$ is a diagonal weight matrix.

Newton-Raphson Algorithm

We solve for β iteratively using:

$$\beta^{(t+1)} = \beta^{(t)} - [\nabla^2 \ell(\beta^{(t)})]^{-1} \nabla \ell(\beta^{(t)})$$

Substituting the gradient and Hessian:

$$\beta^{(t+1)} = \beta^{(t)} + (X^\top W X)^{-1} X^\top (y - \mu)$$

This is equivalent to **Fisher scoring** or **IRLS** used in `glm()`.