第三章笔记

小狗

目录

1 引言

本章我们将学习泊松分布和泊松回归模型。这些概念在处理计数数据时非常重要, 比如研究某段时间内发生的事件次数、每天的顾客数量等。我们会从基础的概率理论开始, 逐步深入到实际应用。

2 矩与累积量

在开始学习泊松分布之前, 我们需要了解一些基本的统计概念: 矩和累积量。这些概念帮助我们描述随机变量的特征。

2.1 矩的定义

矩是描述随机变量分布特征的重要统计量。想象一下, 如果我们把随机变量的所有可能值画在一条线上, 那么矩就是描述这些值如何分布的方法。

第 r 阶矩定义为:

$$m_r = E(Y^r)$$

这里 $E()$ 表示期望值, 也就是平均值。

让我们来看几个重要的矩:

1. 第一阶矩 (m_1): 就是我们常说的平均值或期望值。

$$m_1 = E(Y)$$

2. 第二阶矩 (m_2): 是 Y^2 的平均值。

$$m_2 = E(Y^2)$$

3. 第三阶矩 (m_3): 是 Y^3 的平均值, 用于描述分布的偏斜程度。

$$m_3 = E(Y^3)$$

我们来用 R 计算一下这些矩:

```r
set.seed(123)  # 设置随机种子，确保结果可重复
Y <- rnorm(1000, mean = 10, sd = 2)  # 生成 1000 个均值为 10，标准差为 2 的正态分布随机数

cat(" 第一阶矩 (均值):", mean(Y), "\n")
```

```
## 第一阶矩 (均值): 10.03226
```

```r
cat(" 第二阶矩:", mean(Y^2), "\n")
```

```
## 第二阶矩: 104.5761
```

```r
cat(" 第三阶矩:", mean(Y^3), "\n")
```

```
## 第三阶矩: 1128.494
```

```r
# 计算方差
cat(" 方差:", var(Y), "\n")
```

```
## 方差: 3.933836
```

注意: 方差实际上是基于前两阶矩计算的:$Var(Y) = E(Y^2) - [E(Y)]^2 = m_2 - m_1^2$

2.2 矩生成函数

矩生成函数 (MGF) 是一个强大的工具, 它可以唯一地确定一个分布。MGF 定义为:

$$M(t) = E[\exp(tY)] = \sum_{r=0}^{\infty} \frac{t^r m_r}{r!}$$

这个函数看起来复杂, 但它有一个很好的性质: 我们可以通过对它求导来得到各阶矩。

$$m_r = \frac{d^r M(0)}{dt^r}$$

这意味着, 如果我们知道一个分布的 MGF, 我们就可以计算出它的所有矩!

2.3 累积量生成函数

累积量生成函数 (CGF) 是矩生成函数的自然对数:

$$K(t) = \log(M(t)) = \log(E[\exp(tY)])$$

累积量 κ_r 是 CGF 在 $t = 0$ 处的 r 阶导数:

$$\kappa_r = \frac{d^r K(0)}{dt^r}$$

前几个累积量有特殊的含义:

- $\kappa_1 = E(Y)$: 均值
- $\kappa_2 = Var(Y)$: 方差
- κ_3 用于计算偏度: $Skewness = \frac{\kappa_3}{\kappa_2^{3/2}}$

累积量的一个重要性质是: 独立随机变量的和的累积量等于各个随机变量累积量的和。这在后面学习泊松分布时会很有用。

3 泊松分布

3.1 泊松分布的定义与特征

泊松分布是一种离散概率分布，常用于模拟在固定时间或空间内随机事件发生的次数。例如:

- 一小时内到达商店的顾客数
- 一平方米土地上的植物数量
- 一页书中的印刷错误数

泊松分布的概率质量函数 (PMF) 为:

$$P(Y=k)=\frac{\lambda^k e^{-\lambda}}{k!},\quad k=0,1,2,\ldots$$

这里: - Y 是随机变量 (例如, 事件发生的次数) - k 是具体的次数 - λ 是分布的参数, 表示平均发生率

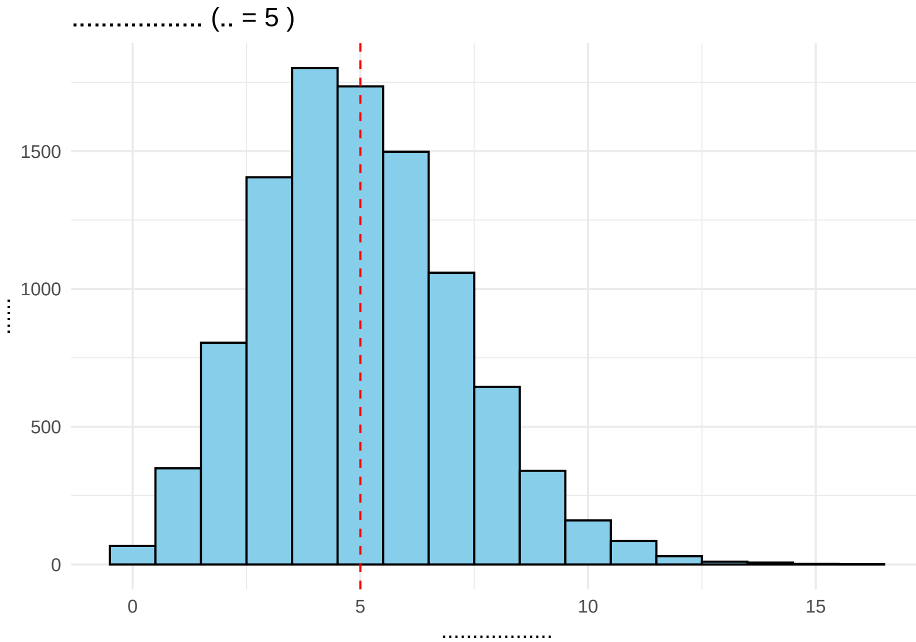
泊松分布的一个重要特性是: 其均值和方差都等于 λ。

$$E(Y)=Var(Y)=\lambda$$

让我们用 R 来模拟泊松分布:

```
# 模拟泊松分布
lambda <- 5
n <- 10000
Y_poisson <- rpois(n, lambda)

# 绘制直方图
ggplot(data.frame(Y = Y_poisson), aes(x = Y)) +
  geom_histogram(binwidth = 1, fill = "skyblue", color = "black") +
  geom_vline(xintercept = lambda, color = "red", linetype = "dashed") +
  labs(title = paste(" 泊松分布模拟 ( =", lambda, ")"),
```

```r
      x = " 事件发生次数", y = " 频率") +
  theme_minimal()
```




```r
# 计算样本均值和方差
cat(" 样本均值:", mean(Y_poisson), "\n")
```

```
## 样本均值: 4.9847
```

```r
cat(" 样本方差:", var(Y_poisson), "\n")
```

```
## 样本方差: 4.896556
```

从图中我们可以看到，当 $\lambda = 5$ 时，泊松分布呈现出一个略微右偏的形状。红色虚线表示均值位置。注意样本均值和方差都接近于 λ。

3.2 泊松分布的矩生成函数和累积量生成函数

泊松分布的矩生成函数为:

$$M(t) = \exp(\lambda(e^t - 1))$$

累积量生成函数为:

$$K(t) = \lambda(e^t - 1)$$

这些函数看起来可能有点复杂, 但它们可以帮助我们轻松计算泊松分布的各阶矩和累积量。

4 泊松回归模型

现在我们来看看如何将泊松分布应用到回归分析中。泊松回归是广义线性模型的一种, 用于分析计数数据。

4.1 模型定义

在泊松回归中, 我们假设响应变量 Y 服从泊松分布, 其均值 μ 与预测变量 X 有关。模型使用对数链接函数:

$$\log(\mu) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$$

或者简洁地写作:

$$\log(\mu) = X^T \beta$$

这里 X 是预测变量的向量,β 是回归系数。

使用对数链接函数确保了 $\mu > 0$, 这符合泊松分布的要求。

4.2 实例: 模拟数据

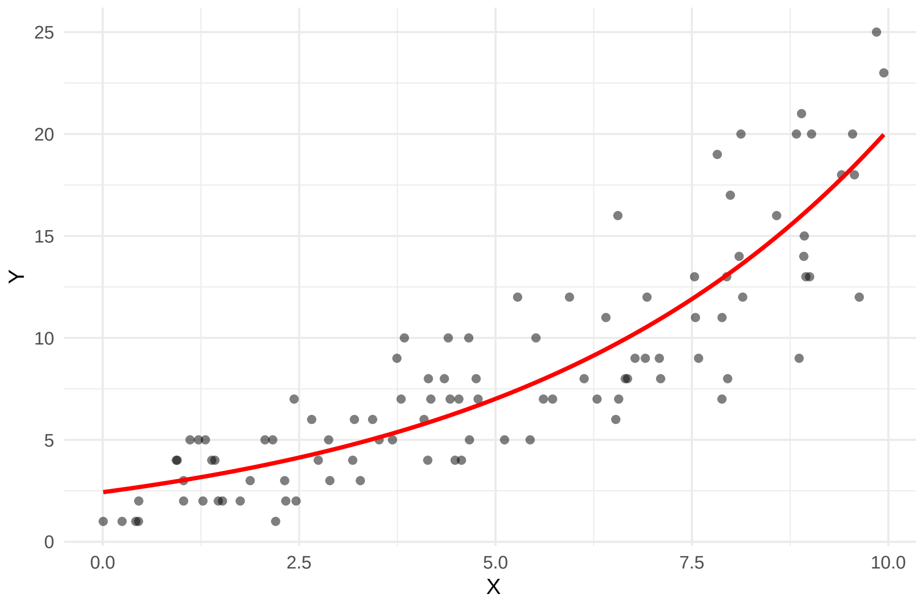
让我们用 R 创建一个简单的泊松回归模型:

```r
set.seed(123)
n <- 100
X <- runif(n, 0, 10)  # 生成 0 到 10 之间的均匀分布随机数作为预测变量
lambda <- exp(1 + 0.2 * X)  # 真实的 lambda 值
Y <- rpois(n, lambda)  # 生成泊松分布的响应变量

# 拟合泊松回归模型
model <- glm(Y ~ X, family = poisson(link = "log"))

# 查看模型摘要
summary(model)
```

```
##
## Call:
## glm(formula = Y ~ X, family = poisson(link = "log"))
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.88879    0.09505   9.351   <2e-16 ***
## X            0.21178    0.01343  15.768   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 353.52  on 99  degrees of freedom
## Residual deviance:  79.36  on 98  degrees of freedom
## AIC: 457.76
##
## Number of Fisher Scoring iterations: 4
```

```
# 绘制数据和拟合线
ggplot(data.frame(X = X, Y = Y), aes(x = X, y = Y)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "glm", method.args = list(family = "poisson"), se = FALSE, color
  labs(title = " 泊松回归模型", x = "X", y = "Y") +
  theme_minimal()
```

....



在这个例子中: - 我们生成了一个预测变量 X 和一个响应变量 Y。- Y 服从泊松分布，其 λ 参数与 X 有关: $\lambda = \exp(1 + 0.2X)$。- 我们使用 `glm()` 函数拟合泊松回归模型。- 模型摘要显示了估计的系数和它们的显著性。- 图表展示了数据点和拟合的回归线。

4.3 解释结果

从模型摘要中，我们可以看到: - 截距估计值接近 1 - X 的系数估计值接近 0.2

这与我们设定的真实模型 $\log(\lambda) = 1 + 0.2X$ 非常接近。

p 值很小表明这些系数在统计上显著。这意味着 X 对 Y 有显著影响。

5 泊松回归率

5.1 泊松过程

泊松过程是一种随时间连续发生的随机事件序列。它有以下特征:

1. 事件以恒定的平均速率 λ 发生。
2. 事件之间的时间间隔服从指数分布。
3. 在不重叠的时间间隔内发生的事件数是独立的。

5.2 偏移项 (Offset) 的使用

在某些情况下, 我们可能需要考虑不同观察单位的”暴露时间”或”风险时间”。这时我们引入偏移项:

$$\log(\mu_i) = \log(t_i) + X_i^T\beta$$

这里 t_i 是第 i 个观察单位的暴露时间。$\log(t_i)$ 作为偏移项加入模型。

5.3 实例: 船只数据

我们来分析 MASS 包中的 ships 数据集, 这个数据集记录了不同类型船只的损坏事故。

```r
data(ships)
# 移除服务时间为 0 的记录
ships <- subset(ships, service > 0)

# 查看数据集的前几行
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
```

```r
# 拟合泊松回归模型
model_ships <- glm(incidents ~ type + offset(log(service)),
                   family = poisson(link = "log"), data = ships)

# 查看模型摘要
summary(model_ships)
```

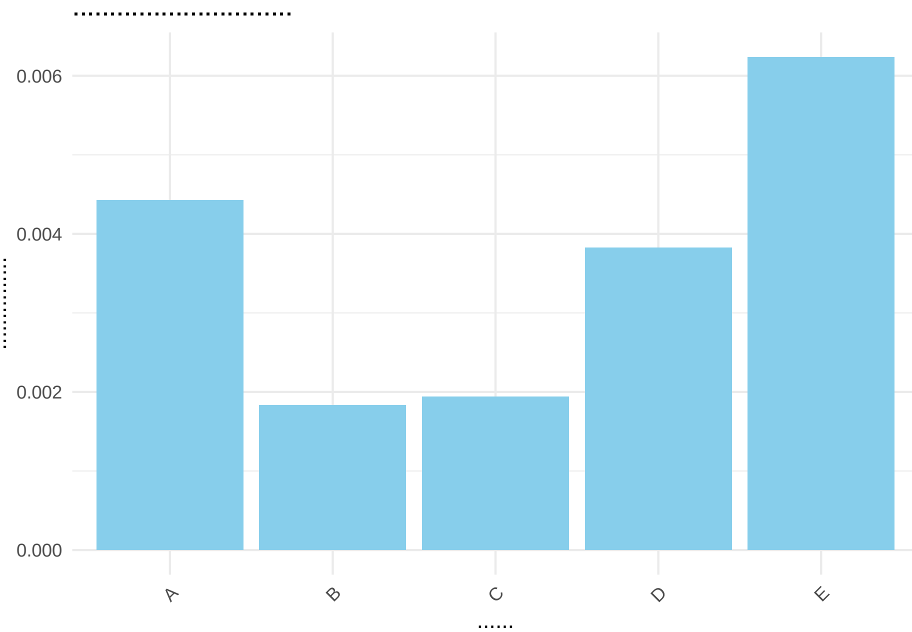
```
##
## Call:
## glm(formula = incidents ~ type + offset(log(service)), family = poisson(link = "log"
##     data = ships)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -5.4202     0.1543 -35.127  < 2e-16 ***
## typeB        -0.8837     0.1666  -5.304 1.13e-07 ***
## typeC        -0.8260     0.3273  -2.524   0.0116 *
## typeD        -0.1459     0.2875  -0.507   0.6118
## typeE         0.3429     0.2346   1.461   0.1439
## ---
## 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:  90.889  on 29  degrees of freedom
```

```
## AIC: 198.76
##
## Number of Fisher Scoring iterations: 5
```

```
# 计算每种船型的事故率 (每个服务月的事故数)
ship_types <- levels(ships$type)
incident_rates <- exp(coef(model_ships)[c("(Intercept)", paste0("type", ship_types[-1])
incident_rates <- c(incident_rates[1], incident_rates[-1] * incident_rates[1])

# 创建数据框来存储结果
results <- data.frame(
  ShipType = ship_types,
  IncidentRate = incident_rates
)

# 打印结果
print(results)
```

```
##             ShipType IncidentRate
## (Intercept)        A  0.004426178
## typeB              B  0.001829132
## typeC              C  0.001937672
## typeD              D  0.003825383
## typeE              E  0.006236601
```

```
# 可视化不同船型的事故率
ggplot(results, aes(x = ShipType, y = IncidentRate)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  labs(title = " 不同船型的每月事故率",
       x = " 船型", y = " 每月事故率") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



解释结果: 1. 模型系数: 每种船型的系数表示相对于 A 型船 (基准水平) 的对数事故率差异。2. 事故率: 我们计算了每种船型每服务月的事故率。3. 图表: 直观地展示了不同船型的事故率差异。

从结果可以看出: - E 型船的事故率最高, 约为每月 0.0062 次事故。- B 型和 C 型船的事故率最低, 约为每月 0.0018 次事故。- A 型和 D 型船的事故率居中。

这种分析可以帮助船运公司了解不同类型船只的风险, 从而制定相应的安全策略和维护计划。

6 结论

泊松回归是处理计数数据的强大工具。它能帮助我们理解预测变量如何影响事件发生的频率。通过使用对数链接函数和偏移项, 泊松回归可以处理各种复杂的情况, 如不同观察时间或暴露程度。

在实际应用中, 泊松回归被广泛用于: - 流行病学研究 (疾病发生率分析) - 生态学 (物种分布和丰度研究) - 质量控制 (产品缺陷分析) - 交通安全 (事故

频率研究) 等多个领域。

理解并正确应用泊松回归, 可以帮助我们从计数数据中获得有价值的洞察和预测。以下是对泊松回归模型的一些进一步讨论和注意事项:

7 泊松回归的进阶话题

7.1 过度离散和欠离散

在实际应用中，我们经常会遇到数据的方差大于均值（过度离散）或小于均值（欠离散）的情况。这违反了泊松分布的假设（均值等于方差）。

7.1.1 过度离散

过度离散是比较常见的情况，可能由以下原因导致：1. 遗漏了重要的预测变量 2. 数据中存在聚集效应 3. 存在异常值

处理过度离散的方法：1. 使用准泊松（Quasi-Poisson）模型 2. 使用负二项回归模型

让我们用 R 来演示如何处理过度离散：

```r
# 模拟过度离散的数据
set.seed(123)
n <- 1000
X <- runif(n, 0, 10)
lambda <- exp(1 + 0.2 * X)
Y <- rnbinom(n, mu = lambda, size = 1)  # 使用负二项分布来模拟过度离散

# 拟合普通泊松回归
poisson_model <- glm(Y ~ X, family = poisson())

# 拟合准泊松回归
quasipoisson_model <- glm(Y ~ X, family = quasipoisson())
```

```r
# 拟合负二项回归
library(MASS)
negbin_model <- glm.nb(Y ~ X)

# 比较模型
AIC(poisson_model, negbin_model)
```

```
##               df       AIC
## poisson_model  2 10824.962
## negbin_model   3  6100.766
```

在这个例子中，我们可以看到负二项模型的 AIC 较低，说明它可能更适合这个过度离散的数据。

7.1.2 欠离散

欠离散虽然不太常见，但在某些情况下也会出现，例如在非常受控的实验中。处理欠离散的一种方法是使用广义泊松模型。

7.2 零膨胀模型

在某些计数数据中，我们可能会观察到过多的零值。例如，研究鱼类数量时，可能有很多地方没有发现鱼。这种情况下，我们可以使用零膨胀泊松模型（ZIP）或零膨胀负二项模型（ZINB）。

```r
# 安装并加载 pscl 包
if (!requireNamespace("pscl", quietly = TRUE)) {
  install.packages("pscl")
}
library(pscl)

# 模拟零膨胀数据
set.seed(123)
```

```r
n <- 1000
X <- runif(n, 0, 10)
lambda <- exp(1 + 0.2 * X)
zero_prob <- 0.3
Y <- ifelse(runif(n) < zero_prob, 0, rpois(n, lambda))

# 拟合零膨胀泊松模型
zip_model <- zeroinfl(Y ~ X | X, dist = "poisson")

summary(zip_model)
```

```
##
## Call:
## zeroinfl(formula = Y ~ X | X, dist = "poisson")
##
## Pearson residuals:
##     Min      1Q  Median      3Q     Max
## -1.2980 -1.1993  0.1504  0.7768  3.2581
##
## Count model coefficients (poisson with log link):
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 1.042211   0.034465   30.24   <2e-16 ***
## X           0.194060   0.004932   39.35   <2e-16 ***
##
## Zero-inflation model coefficients (binomial with logit link):
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.12109    0.14853  -7.548 4.42e-14 ***
## X            0.04957    0.02492   1.990   0.0466 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of iterations in BFGS optimization: 10
## Log-likelihood: -2281 on 4 Df
```

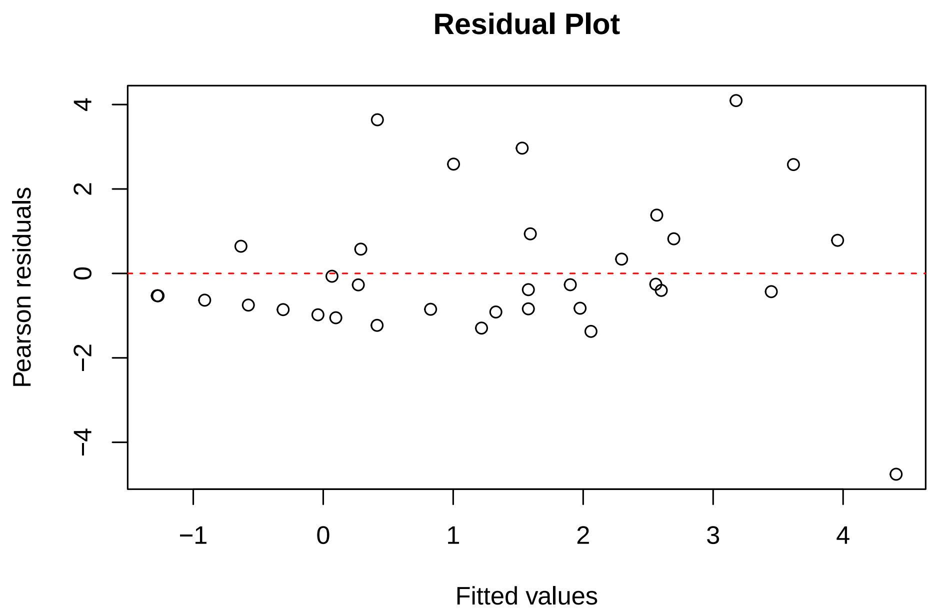
在这个模型中，我们同时建模了零的概率和非零计数的期望值。

8 泊松回归的诊断和模型评估

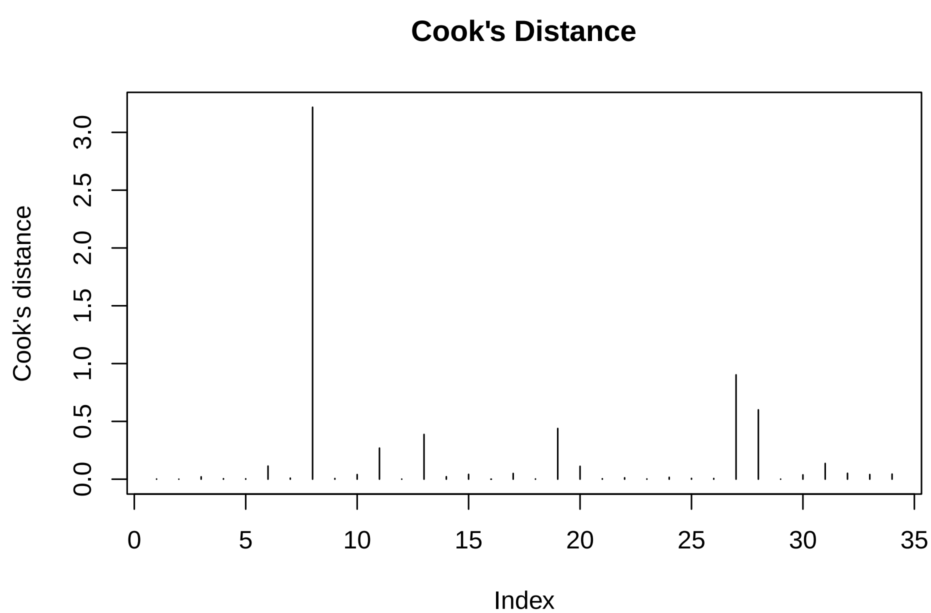
在使用泊松回归模型后，我们需要进行一些诊断检查，以确保模型假设得到满足：

1. 残差分析
2. 影响点检测
3. 多重共线性检查

```
# 使用之前的 ships 数据
model <- glm(incidents ~ type + offset(log(service)),
             family = poisson(link = "log"), data = ships)

# 残差图
plot(predict(model), residuals(model, type = "pearson"),
     xlab = "Fitted values", ylab = "Pearson residuals",
     main = "Residual Plot")
abline(h = 0, col = "red", lty = 2)
```




```r
# 影响点检测
plot(cooks.distance(model), type = "h",
     main = "Cook's Distance", ylab = "Cook's distance")
```



这些图可以帮助我们识别潜在的问题，如异常值或影响点。

9 泊松回归在实际中的应用

泊松回归在多个领域都有广泛应用，以下是一些具体例子：

1. 公共卫生：分析疾病发病率，预测医院就诊人数。
2. 生态学：研究物种分布和丰度。
3. 交通安全：分析交通事故频率。
4. 质量控制：研究产品缺陷出现的频率。
5. 犯罪学：分析犯罪发生率。

例如，在疫情期间，泊松回归可以用来分析影响 COVID-19 传播的因素：

```r
# 模拟 COVID-19 数据
set.seed(123)
n <- 100
population_density <- runif(n, 10, 1000)
vaccination_rate <- runif(n, 0.3, 0.9)
cases <- rpois(n, lambda = exp(2 + 0.001 * population_density - 3 * vaccination_rate))

covid_data <- data.frame(cases, population_density, vaccination_rate)

# 拟合泊松回归模型
covid_model <- glm(cases ~ population_density + vaccination_rate,
                   family = poisson(), data = covid_data)

summary(covid_model)
```

```
##
## Call:
## glm(formula = cases ~ population_density + vaccination_rate,
##     family = poisson(), data = covid_data)
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)         2.3245247  0.3013504   7.714 1.22e-14 ***
```

```
## population_density  0.0008232  0.0002550   3.228  0.00125 **
## vaccination_rate   -3.4561523  0.4749204  -7.277 3.40e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 188.77  on 99  degrees of freedom
## Residual deviance: 116.41  on 97  degrees of freedom
## AIC: 342.45
##
## Number of Fisher Scoring iterations: 5
```

这个模型可以帮助我们理解人口密度和疫苗接种率如何影响 COVID-19 病例数。

10 结论

泊松回归是一个强大的统计工具，适用于各种计数数据分析。它的优势在于:

1. 能够处理非负整数响应变量
2. 可以纳入多个预测变量
3. 可以通过对数链接函数建立非线性关系
4. 可以通过偏移项处理暴露时间或率的问题

然而，使用泊松回归时也需要注意一些潜在的问题，如过度离散、零膨胀等。通过适当的诊断和必要的模型调整，我们可以确保得到可靠的结果。

随着数据科学和机器学习的发展，泊松回归也在不断演化。例如，泊松回归树、泊松随机森林等方法将泊松回归与更复杂的算法结合，以处理非线性关系和高维数据。

总的来说，掌握泊松回归及其相关概念，对于处理各种计数数据问题都是非常有价值的。它不仅是统计学中的重要工具，也是数据科学家和研究人员应

该熟悉的重要方法之一。