考试题型包括问答和上机两部分：  
问答部分（不需要写代码）：45分  
（1）判断 True or False（3个，共9分）  
（2）单向选择Multiple-Choice（5个，共15分）  
（3）简答Short answer（1个，21分）  
知识点：  
线性/广义线性回归的前提假设有哪些。  
根据数据描述，选择合适的拟合模型。（线性，泊松，二项）  
广义线性回归的链接函数分别是什么。  
广义线性回归会写拟合公式，会解释拟合公式。  
会解读自变量系数，掌握自变量变化对响应变量的影响，会计算和解读置信区间。会根据链接函数还原响应变量的变化区间。  
quasipoisson什么时候用，结果解释。  
如何检验广义线性回归数据是否符合假设的分布，解释结果。 1 - pchisq（）  
几种残差的适用选择。  
会根据模型似然值计算AIC/BIC，AIC/BIC大小代表什么意思，AIC, AICc 和 BIC差别，AIC适用模型。  
会根据GAM plot分析是否存在非线性关系。  
读ROC Curves写出sensitivity (true positive rate) and specificity (true negative rate)，计算混淆矩阵。

线性回归和广义线性回归（GLM）各自有一系列前提假设，确保模型能够有效地进行参数估计和推断。以下是这两种回归模型的前提假设：

1. 线性回归 (Linear Regression) 的前提假设

线性回归是最基础的回归分析方法，其主要假设包括：

1.1. 线性关系 (Linearity)

假设自变量 XX 和因变量 YY 之间存在线性关系，即回归方程的形式为：

1.2. 独立性 (Independence)

假设各个观测值之间是独立的，即一个观测值的误差项不应该影响其他观测值的误差项。这是指自变量之间和因变量之间没有相关性。

1.3. 误差项的正态性 (Normality of Errors)

假设误差项 ϵ\epsilon 服从正态分布，特别是在小样本情况下，正态性假设确保了估计量的无偏性和有效性。

1.4. 同方差性 (Homoscedasticity)

假设误差项的方差是恒定的，不随自变量的取值或因变量的变化而变化。如果误差项的方差随着自变量或因变量的变化而变化，称为异方差性 (Heteroscedasticity)。

1.5. 无多重共线性 (No Multicollinearity)

假设自变量之间不存在高度相关性。多重共线性会导致回归系数的不稳定性，从而影响模型的推断。

1.6. 误差项的独立同分布 (IID)

误差项应该是独立同分布的，即误差项之间不应该有序列相关，且各个误差项的分布应该相同。

2. 广义线性回归 (Generalized Linear Models, GLM) 的前提假设

广义线性模型是对线性回归的扩展，能够处理不同类型的因变量（如二项分布、泊松分布等）。GLM的前提假设相对灵活，包含以下几项：

2.1. 线性预测 (Linearity in the Linear Predictor)

假设因变量的对数（或其他适当的变换）与自变量之间存在线性关系。对于GLM，通常通过链接函数（link function）来描述：

2.2. 误差分布 (Distribution of Errors)

假设因变量 YY 来自于某个特定的分布族（例如，二项分布、泊松分布等）。这意味着数据应该符合某个概率分布，通常采用指数族分布：

2.3. 独立性 (Independence)

与线性回归类似，GLM假设不同观测值之间是独立的。每个观测值的误差项不应与其他观测值相关。

2.4. 链接函数的适用性 (Link Function Suitability)

选择合适的链接函数（如对数链接、逻辑链接等）是GLM的一个关键前提。链接函数需要能够将线性预测量与期望值之间的关系建立起来。

2.5. 无多重共线性 (No Multicollinearity)

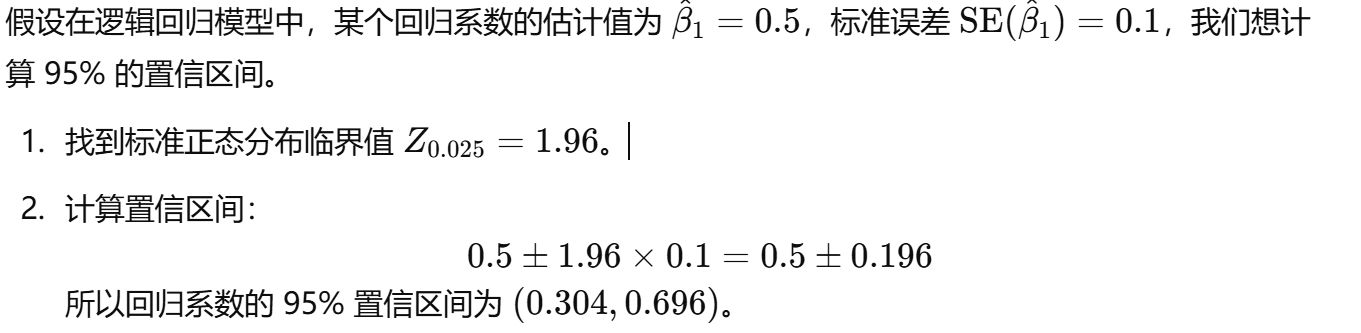
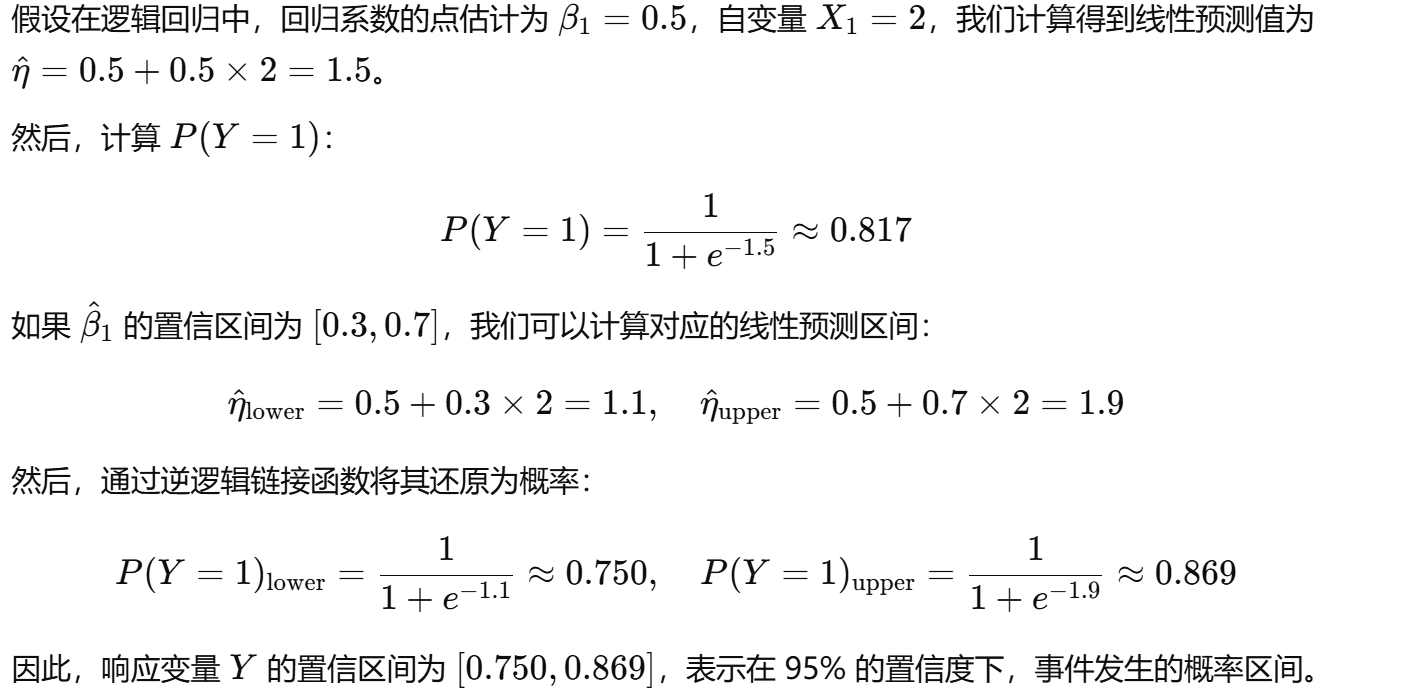
类似于线性回归，GLM也要求自变量之间没有高度的多重共线性，以确保模型估计的稳定性。

3. 总结对比

* 线性回归 假设因变量和自变量之间存在线性关系，误差项独立同分布，且有恒定方差等。
* 广义线性回归 则通过选择不同的分布族（如二项、泊松等）和链接函数，适应更广泛的应用场景，例如处理非正态分布的因变量。

这些前提假设对回归模型的有效性和推断至关重要，违背这些假设可能导致模型估计不准确或不可靠。

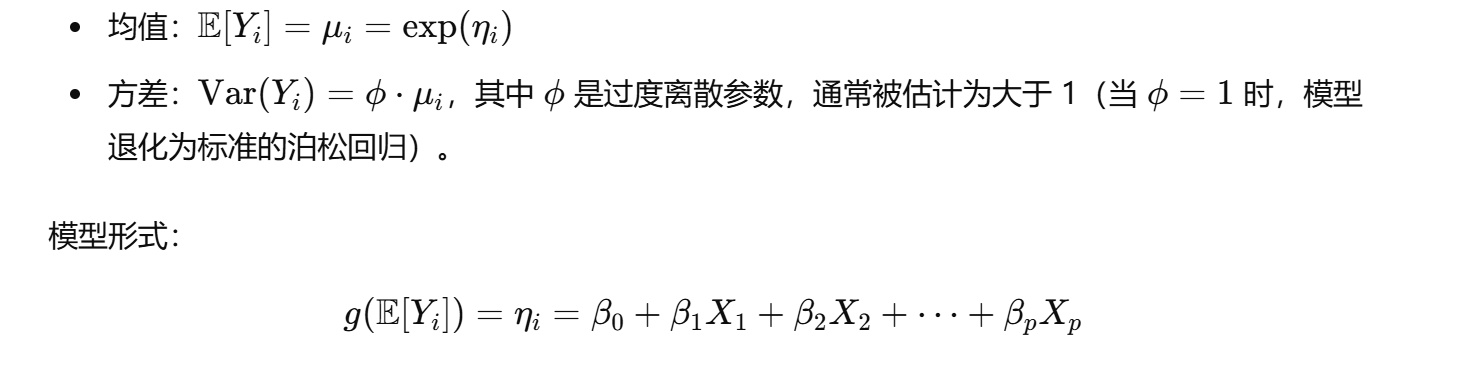


**1. 应用场景**

**泊松回归**模型适用于计数数据建模，如某时间段内发生的事件数量（例如疾病发生的次数、事故发生的次数等）。假设数据符合泊松分布，**即均值和方差相等。**

然而，在实际数据中，尤其是在计数数据中，往往会遇到 **过度离散**（overdispersion）问题，即观察到的方差大于均值。这时，传统的泊松回归不再适用，因为它假设方差等于均值。

**Quasi-Poisson** 是一种改进方法，它通过引入一个额外的过度离散参数来调整方差，使其不再局限于均值。  Since Quasi-Poisson adjusts for overdispersion, it helps correct the **standard errors** and thus provides more reliable **p-values** for testing the significance of the regression coefficients.

 Without accounting for overdispersion, Poisson regression could underestimate standard errors and lead to **Type I errors** (wrongly rejecting a true null hypothesis). Quasi-Poisson regression mitigates this problem by properly adjusting the standard errors.

以下是对 **普通残差 (Raw Residuals)**、**皮尔逊残差 (Pearson Residuals)** 和 **偏差残差 (Deviance Residuals)** 的详细描述，并附上专业术语的英文注释：

### 1. **普通残差 (Raw Residuals)**

**适用场景**：

* **适合用于简单回归模型** (Simple Linear Regression) 或者 **没有严重异方差性和离群点的数据** (data without severe heteroscedasticity and outliers)。
* 用于 **基础模型诊断** (basic model diagnostics)，例如检查拟合的准确性。

**特点**：

* 计算简单直观，直接显示预测误差。
* 在数据集不存在异方差性或离群点的情况下，普通残差能够有效地反映模型拟合误差。

### 2. **皮尔逊残差 (Pearson Residuals)**

**适用场景**：

* **适用于广义线性回归（GLM）** (Generalized Linear Models)，特别是当数据具有 **异方差性** (heteroscedasticity) 的情况下。
* 用于 **标准化普通残差** (standardizing raw residuals)，能够比较不同数据点的拟合误差。

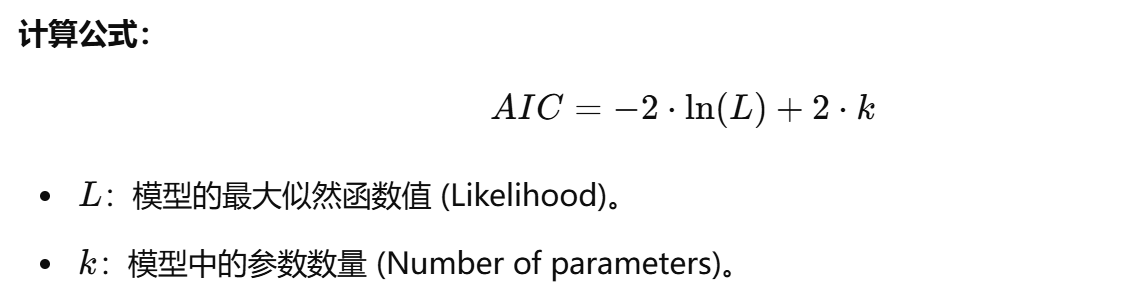
**特点**：

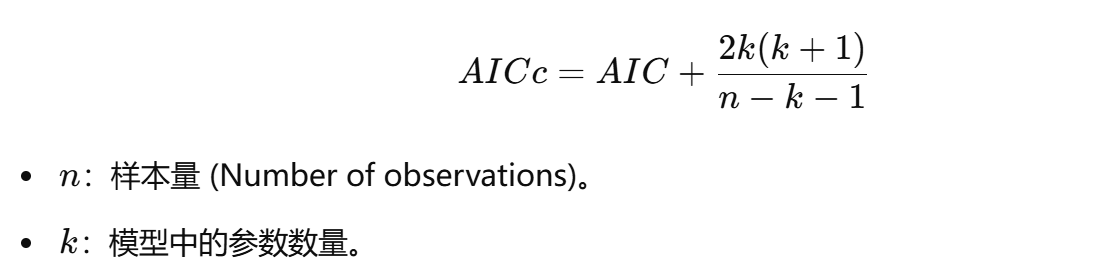
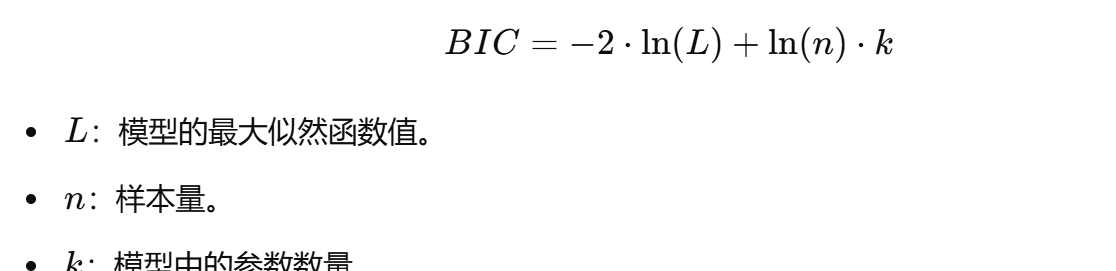
* 通过标准化，皮尔逊残差消除了每个数据点的方差差异。
* 适用于具有异方差性的数据，尤其是在 **泊松回归** (Poisson Regression) 或 **二项回归** (Binomial Regression) 中常见。

### 3. **偏差残差 (Deviance Residuals)**

**适用场景**：

* **适用于广义线性回归（GLM）** 模型，尤其是在数据符合 **非正态分布** (non-normal distributions) 的情况下，如 **泊松回归** (Poisson Regression) 或 **二项回归** (Binomial Regression)。
* 用于 **评估模型拟合优度** (assessing model fit) 和 **诊断模型拟合不良** (diagnosing poor model fit)。
* 通过对数似然差异来量化模型的拟合优度，偏差残差能够准确诊断 **非正态分布** 数据的拟合效果。
* 在 **泊松回归** 或 **二项回归** 等非正态分布模型中，偏差残差用于进一步评估模型的拟合质量和改进模型的需要。



 **AIC**: A smaller value indicates that the model has better predictive performance while fitting the data.

**BIC**: A smaller value indicates that the model is better at fitting the data while balancing complexity.

**AIC**：适用于多种模型，尤其在预测能力优先时。AIC 越小越好

 适用于 **广义线性模型 (GLM)**、**时间序列模型**、**回归模型** 等多种统计模型。

 假设模型的误差是独立同分布的，但不强制要求特定的分布。

会根据GAM plot分析是否存在非线性关系。

 **存在非线性关系**：GAM plot 中平滑函数的曲线明显偏离线性趋势。

 **没有非线性关系**：平滑函数接近直线或水平线。 通过 GAM plot 的形状和置信区间，可以直接判断变量与响应变量之间是否存在非线性关系，从而决定模型的调整策略。

上机部分（需要写代码）：55分  
知识点

绘制散点图，不同类别的散点用颜色区分。

plot(x, y, type = "p", main = "Title", xlab = "X-axis", ylab = "Y-axis", col = "blue", pch = 16)

plot(x, y1, col = "blue", pch = 16, xlab = "X-axis", ylab = "Y-axis", main = "Scatter Plot with Groups")

points(x, y2, col = "red", pch = 17) # 添加第二组散点

# 添加图例 legend("topleft", legend = c("Group1", "Group2"), col = c("blue", "red"), pch = c(16, 17))

会使用二项分布/泊松分布/ Quasi-binomial/ Quasi-poisson拟合数据，解读模型的dispersion参数的含义。

quasi\_binom\_fit <- glm(y\_binom ~ x, family = quasibinomial(link = "logit"), data = data) summary(quasi\_binom\_fit)

quasi\_pois\_fit <- glm(y\_pois ~ x, family = quasipoisson(link = "log"), data = data) summary(quasi\_pois\_fit)  
绘制拟合模型残差分布图。

#绘制residual曲线

predicted(model.fit)

residuals(model.fit)

plot(predicted,residuals)

//special residuals

residuals(model.fit,type="response")

residuals(model.fit,type="Pearson")

residuals(model.fit,type="Deviance")

#抖动的残差 适合离散数据

library(statmod)

qresiduals(model.fit)//randomised quantile residual

#绘制残差图

plot(predict(model.fit),residual(model.fit,type="??"))

abline(h=0,lty="dashed")

plot(model.fit,which=1)   
会使用anova函数分析分类变量及交互项是否显著。

Anova(model, test = “Chisq”)  
会绘制几种残差图。  
掌握使用parametric bootstrap simulation和nonparametric bootstrap simulation估算参数和其置信区间。

set.seed(12345)

beta0.true = 0.5

beta1.true = -1.2

sample.size = 1000

n.sim = 123

parameters.func = function(beta0.true,beta1.true,sample.size){

  x = seq(0, 1, length.out = sample.size)

  exp.true = exp(beta0.true + beta1.true\*x)

  pi.true = exp.true/(1+exp.true)

  list(sample.size=sample.size,x=x,pi.true=pi.true)

}

sim.logistic.func = function(para.list){

  sample.size=para.list$sample.size

  x=para.list$x

  pi.true=para.list$pi.true

  y = rbinom(n=sample.size,size=1,prob=pi.true)

  sim.logistic.df = data.frame(y=y,x=x)

  logistic.fit = glm(y~x, family = "binomial", data = sim.logistic.df)

  return(logistic.fit)

}

dev.vec = double(n.sim)

for (i in 1:n.sim){

  logistic.fit = sim.logistic.func(

    parameters.func(

      beta0.true=beta0.true,beta1.true=beta1.true,

      sample.size=sample.size))

  dev.vec[i] = deviance(logistic.fit)

}

mean(dev.vec>1369)

**答案解析：**

# Add some code here

n.sims=10000

## Creating vector in which to store estimates.

bstrap.medians=bstrap.means=double(n.sims)

## The for loop.

for (i in 1:n.sims) {

  sam=sample(1:n,replace=TRUE)

  bstrap.y=y[sam]

  bstrap.medians[i]=median(bstrap.y)

  bstrap.means[i]=mean(bstrap.y)

}

#Non-parametric bootstrap CI for medians

quantile(bstrap.medians,c(.025,.975))

#Non-parametric bootstrap CI for means

quantile(bstrap.means,c(.025,.975))

# These confidence intervals contain the true values.

```

 2.5%     97.5%

0.8021284 1.3546304

    2.5%    97.5%

1.337267 2.013136

会通过GAM观察是否需要增加自变量的二次项。

library(mgcv)

library(VGAM)

> fit1a = gam(y ~ s(x1) + s(x2) + x3, binomial, bdata) # mgcv

> fit1b = gam(y ~ s(x1) + s(x2, df = 1) + x3, binomial, bdata) # gam

> fit2 = vgam(y ~ s(x1) + s(x2, df = 1) + x3, binomialff, bdata) # VGA  
会使用dredge做model selection，会读取拟合的model。

library(MuMIn)

step(model.fit,direction="backward")

step(model.fit,direction="forward")

step(model.fit,direction="both")//the default criterion is AIC

evap.fits <- dredge(evap.lm)

print(round(evap.fits[1:10, ], 2))

options(na.action = "na.fail", width=120)

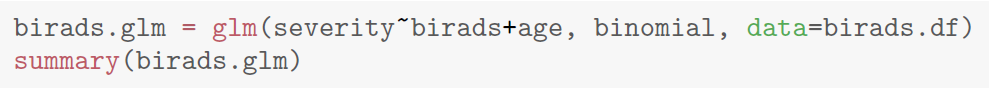
evap2.fits <- dredge(evap.lm, rank="BIC")

print(round(evap2.fits[1:10, ], 2))

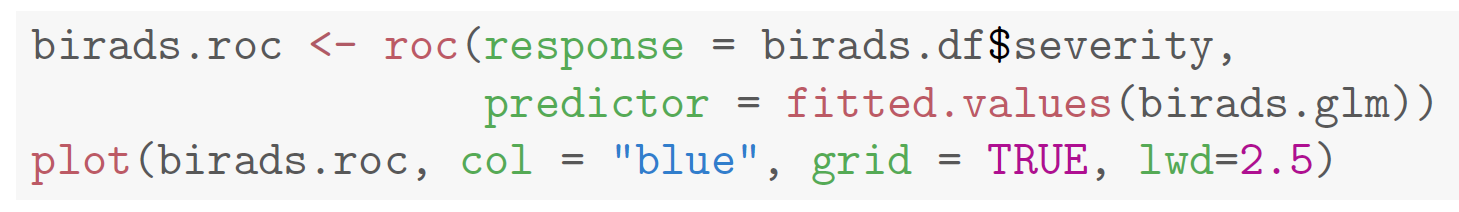
model1.lm = get.models(evap.fits, 1)[[1]]

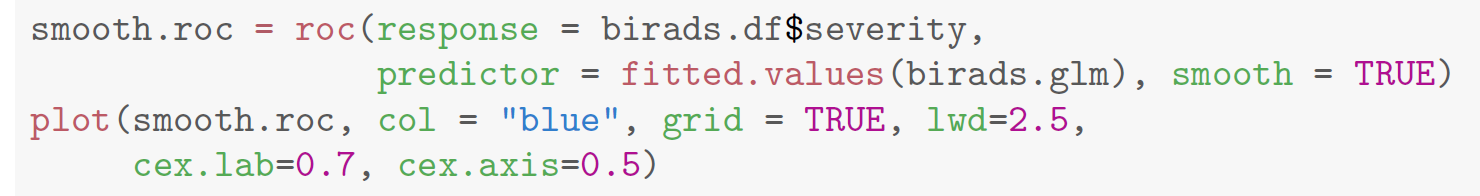
summary(model1.lm)

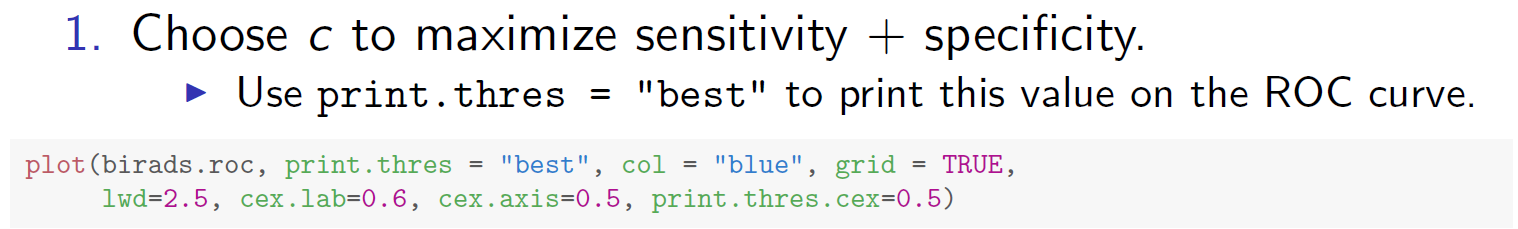
会获取拟合模型的ROC曲线，并计算相关的参数：Auc、Sensitivity、 Specificity、 Prediction error。











library(pROC)

handwriting.df$gender=ifelse(handwriting.df$gender=="F", yes = 0, no = 1)

handwriting.test.df$gender=ifelse(handwriting.test.df$gender=="F", yes = 0, no = 1)

pi.est = predict(full.fit, newdata = handwriting.test.df, type = "response")

y.est = as.numeric(pi.est>0.5)

obj.tab = table(actual = handwriting.test.df$gender, pred = y.est)

obj.tab[2,2] / sum(obj.tab[2,]) # Sensitivity

obj.tab[1,1] / sum(obj.tab[1,]) # Specificity

total = sum(obj.tab)

error = total - sum(diag(obj.tab)) # Prediction error

error/total

pi.est = predict(full.fit, newdata = handwriting.test.df, type = "response")

full.test.roc = roc(response = handwriting.test.df$gender,

                 predictor = pi.est)

index.test = which.max(full.test.roc$sensitivities + full.test.roc$specificities)

full.test.roc$thresholds[index.test]

full.test.roc$sensitivities[index.test]

full.test.roc$specificities[index.test]

index.test = which.min(abs(full.test.roc$sensitivities - full.test.roc$specificities))

full.test.roc$thresholds[index.test]

full.test.roc$sensitivities[index.test]

full.test.roc$specificities[index.test]

abs(full.test.roc$auc - full.roc$auc)