《属性数据分析》代码

2018-09-05

目录

表格

插图

前言

这个文档是《属性数据分析》第二版[1]（An Introduction to Categorical Data Analysis, Second Edition[2] 书上部分案例与习题的 R 实现。

以下是对本文档的一些说明：

1. 文档另外配套了 R 包 `cdabookdb` 和 `cdabookfunc`，这些 R 包中包含了教材中会用到的数据与一些用得到的函数。该包的安装和使用说明请看附录 A。
2. 每个案例中引用的数据集均可在 `cdabookdb` 包中找到。文档中的案例用到的全部数据以及教材中所有习题的数据在该包中数据集名称列表可查看附录 B。
3. 文档中每个案例都是独立的，也就是说后面的案例的结果并不会利用到前面计算得到的结果或载入的包等。
4. 文档中章节号与教材保持一致，章节标题与中文教材保持一致，因此若教材中对应章节没有需要 R 实现的案例，则本文档中该章节内容为空。
5. 文档为多人合作完成，代码风格与描述风格等会有一定差异。
6. 文档目前已完成前六章的案例的 R 实现，`cdabookdb` 包目前已完成前七章全部数据的录入。
7. 文档提供了多种格式，可以从网页版 (gitbook 版) 顶端的下载按钮处下载。分别为 pdf 版、equb3 版、zip 版（gitbook 版文件的压缩版）。

[1] http://item.jd.com/10000214.html

[2] https://onlinelibrary.wiley.com/doi/book/10.1002/0470114754

第一章　导言

1.1　属性响应数据

1.2　属性数据的概率分布

二项分布计算

```
# 二项分布概率的计算
dbinom(0, 10, 0.2)  # 10 次试验，每次成功概率 0.2，成功 0 次
```

```
## [1] 0.1074
```

```
# 给定参数的情况下批量累积概率
n <- 10
prob_matrix <- sapply(c(0.2, 0.5, 0.8), function(p) pbinom(0:n, n, p))
dimnames(prob_matrix) <- list(0:n, c("P=0.2", "P=0.5", "P=0.8"))
xtable::xtable(prob_matrix, align = "cccc", digits = 3)
```

P=0.2	P=0.5	P=0.8
0.107	0.001	0.000
0.376	0.011	0.000
0.678	0.055	0.000
0.879	0.172	0.001
0.967	0.377	0.006
0.994	0.623	0.033
0.999	0.828	0.121
1.000	0.945	0.322
1.000	0.989	0.624
1.000	0.999	0.893
1.000	1.000	1.000

```
# 给定参数的二项分布的均值和标准差
n <- 10
p <- 0.2
n * p  # 均值
```

```
## [1] 2
```

```
sqrt(n * p * (1 - p))  # 标准差
```

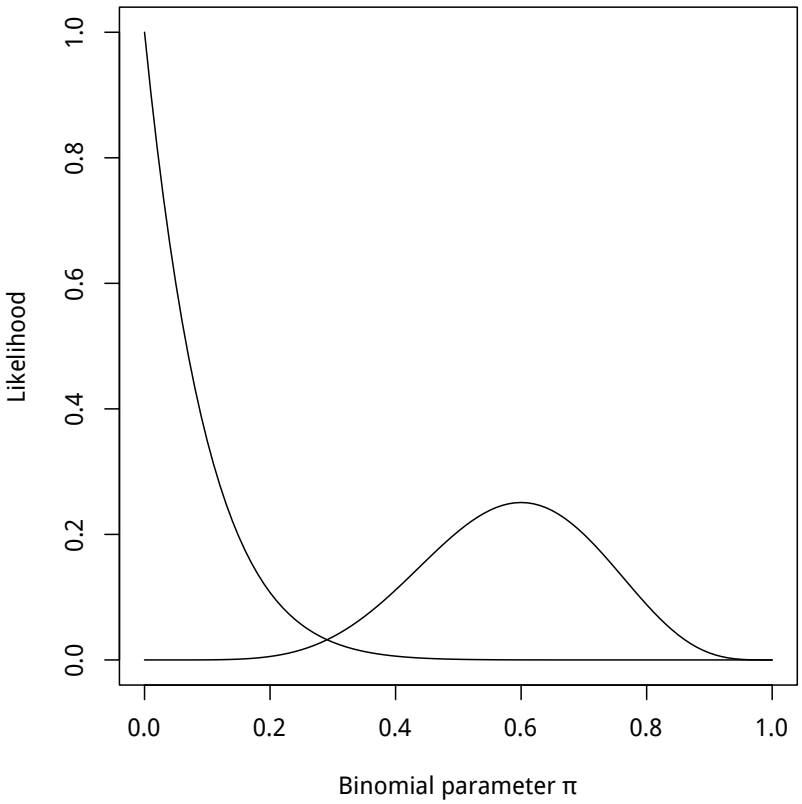
```
## [1] 1.265
```

1.3 比例的统计推断

二项分布似然函数图

书本上的图 1.1

```
prob <- seq(0, 1, 0.01)
prob_plot_data <- data.frame(
  Prob = prob,
  Y_0 = dbinom(0, 10, prob),
  Y_6 = dbinom(6, 10, prob)
)

par(pty = "s")
plot(
  Y_0 ~ Prob, type = "l",
  data = prob_plot_data,
  asp = 1,
  xlab = "Binomial parameter  ",
  ylab = "Likelihood"
)
lines(Y_6 ~ Prob, type = "l", data = prob_plot_data)
```



二项分布假设检验

二项分布的检验分为两种，以下例子使用的数据来自 1.3.3 节的堕胎合法化调查

一种是精确的二项检验，使用 `binom.test()`

```
binom.test(400, 893)
```

```
##
##  Exact binomial test
##
## data:  400 and 893
## number of successes = 400, number of trials = 890,
## p-value = 0.002
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.4150 0.4812
## sample estimates:
## probability of success
##                 0.4479
```

另一种是正态（或卡方）近似的二项检验，可使用 `prob.test()`

```
prop.test(400, 893)
```

```
##
##  1-sample proportions test with continuity correction
##
## data:  400 out of 893, null probability 0.5
## X-squared = 9.5, df = 1, p-value = 0.002
## alternative hypothesis: true p is not equal to 0.5
## 95 percent confidence interval:
##  0.4151 0.4813
## sample estimates:
##      p
## 0.4479
```

```
# correst=FALSE 表示不做连续性调整
prop.test(400, 893, correct = FALSE)
```

```
##
##  1-sample proportions test without continuity
##  correction
##
## data:  400 out of 893, null probability 0.5
## X-squared = 9.7, df = 1, p-value = 0.002
## alternative hypothesis: true p is not equal to 0.5
## 95 percent confidence interval:
##  0.4156 0.4807
## sample estimates:
##      p
## 0.4479
```

1.3.2 节和 1.3.3 节介绍和使用的是未经连续性调整的大样本近似

三个检验的 p 值都小于 0.05，从而拒绝原假设

二项分布置信区间

上一部分二项分布假设检验输出的结果中已包含置信区间

其中 `prop.test(correct = FALSE)` 输出的是书中介绍的第一种调整方法计算的置信区间

```
prop.test(9, 10, 0.9, correct = FALSE)$conf.int
```

```
## Warning in prop.test(9, 10, 0.9, correct = FALSE): Chi-
```

```
## squared approximation may be incorrect
```

```
## [1] 0.5958 0.9821
## attr(,"conf.level")
## [1] 0.95
```

而对于第二种调整方法，也就是 Agresti–Coull confidence interval，R 没有自带的函数可以计算，但可以通过 `binom` 包中的 `binom.agresti.coull()` 函数计算（同时，也可以使用 `binom.confint()` 函数计算多种置信区间的汇总表）

```
library(binom)
binom.agresti.coull(9, 10)
```

```
##          method x  n mean lower upper
## 1 agresti-coull 9 10  0.9 0.574 1.004
```

```
binom.confint(9, 10)
```

```
##           method x  n   mean  lower  upper
## 1  agresti-coull 9 10 0.9000 0.5740 1.0039
## 2     asymptotic 9 10 0.9000 0.7141 1.0859
## 3          bayes 9 10 0.8636 0.6692 0.9996
## 4        cloglog 9 10 0.9000 0.4730 0.9853
## 5          exact 9 10 0.9000 0.5550 0.9975
## 6          logit 9 10 0.9000 0.5328 0.9861
## 7         probit 9 10 0.9000 0.5879 0.9904
## 8        profile 9 10 0.9000 0.6283 0.9904
## 9            lrt 9 10 0.9000 0.6284 0.9940
## 10     prop.test 9 10 0.9000 0.5412 0.9948
## 11        wilson 9 10 0.9000 0.5958 0.9821
```

1.4 关于离散数据的更多统计推断

二项分布参数统计推断

对于 Wald, Score, and Likelihood-Ratio 这三种推断方法

```
# 参数设定
p <- 0.9
n <- 10
pi <- 0.5
```

```
# Wald test
SE <- sqrt(p * (1 - p) / n)
z <- (p - pi) / SE; z
```

```
## [1] 4.216
```

```
# Score test
SE <- sqrt(pi * (1 - pi) / n)
z <- (p - pi) / SE; z
```

```
## [1] 2.53
```

```
# likelihood-ratio test
x <- n * p
L0 <- dbinom(x, n, pi)
L1 <- dbinom(x, n, p)
z <- -2 * log(L0 / L1); z
```

```
## [1] 7.361
```

或者可以使用 cdabookcode 中定义的 binom_inference() 函数计算

```
library(cdabookfunc)
binom_inference(0.9, 10, 0.5, method = "wald")
```

```
## $z
## [1] 4.216
##
## $method
## [1] "wald"
```

```
binom_inference(0.9, 10, 0.5, method = "l")
```

```
## $z
## [1] 7.361
##
## $method
## [1] "likelihood-ratio test"
```

小样本推断

```
# one-side test pvalue
# (H0: pi = 0.5) vs (H1: pi > 0.5)
# p-value = P(Y >= 9) = P(Y > 8)
1 - pbinom(8, 10, 0.5)
```

```
## [1] 0.01074
```

```
# two-side test pvalue
# (H0: pi = 0.5) vs (H1: pi != 0.5)
# p-value = 1 + P(Y <= 1) + P(Y >= 9) = 2 * P(Y > 8)
pbinom(1, 10, 0.5) + pbinom(8, 10, 0.5, lower.tail = FALSE)
```

```
## [1] 0.02148
```

```
2 * (1 - pbinom(8, 10, 0.5))
```

```
## [1] 0.02148
```

小样本推断 P 值调整

小样本推断是保守的，可以使用经过调整的 p 值

中点 P 值可使用 `binom_mid_pvalue()` 计算

```
library(cdabookfunc)
binom_mid_pvalue(9, 10, "g")  # right-tail p-value
```

```
## $pvalue
## [1] 0.005859
##
## $alternative
## [1] "greater"
```

```
binom_mid_pvalue(9, 10)  # two-sided p-value
```

```
## $pvalue
## [1] 0.01172
##
## $alternative
## [1] "two.sided"
```

```
# 获取表 1.2
pvalue_matrix <- cbind(
  0:10,
  dbinom(0:10, 10, 0.5),
  1 - pbinom(-1:9, 10, 0.5),
  binom_mid_pvalue(0:10, 10, "g")$pvalue
)
dimnames(pvalue_matrix) <- list(0:10, c("y", "P(y)", "P-value", "Mid P-value"))
xtable::xtable(pvalue_matrix, align = "ccccc", digits = c(0, 0, 4, 4, 4))
```

y	P(y)	P-value	Mid P-value
0	0.0010	1.0000	0.9995
1	0.0098	0.9990	0.9941
2	0.0439	0.9893	0.9673
3	0.1172	0.9453	0.8867
4	0.2051	0.8281	0.7256
5	0.2461	0.6230	0.5000
6	0.2051	0.3770	0.2744
7	0.1172	0.1719	0.1133
8	0.0439	0.0547	0.0327
9	0.0098	0.0107	0.0059
10	0.0010	0.0010	0.0005

课后题

第 4 题

(a)

```
# (a)
pi <- 0.5

result <- dbinom(0:2, 2, 0.5)
names(result) <- paste0("P(Y=", 0:2, ")")
result
```

```
## P(Y=0) P(Y=1) P(Y=2)
##   0.25   0.50   0.25
```

```
2 * 0.5  # 均值
```

```
## [1] 1
```

```
sqrt(2 * 0.5 * 0.5)  # 标准差
```

```
## [1] 0.7071
```

(b)

```
# (b)(i)
dbinom(0:2, 2, 0.6)
```

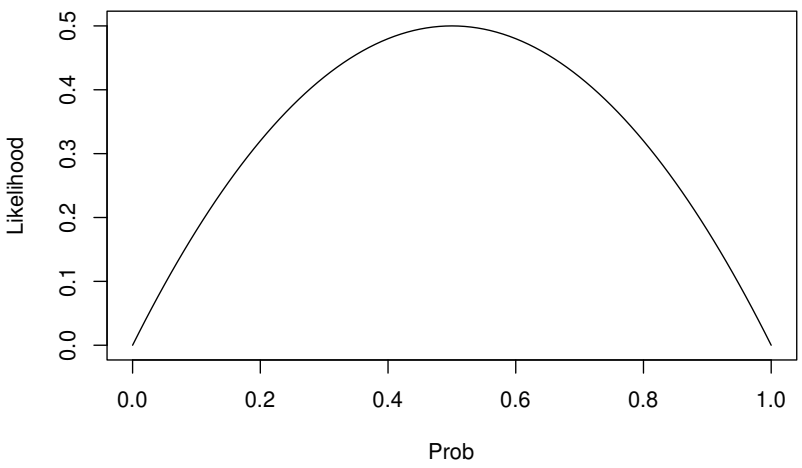
```
## [1] 0.16 0.48 0.36
```

```
# (b)(ii)
dbinom(0:2, 2, 0.4)
```

```
## [1] 0.36 0.48 0.16
```

(c)

```
prob <- seq(0, 1, 0.01)
prob_plot_data <- data.frame(
  Prob = prob,
  Y_1 = dbinom(1, 2, prob)
)

plot(
  Y_1 ~ Prob, type = "l",
  data = prob_plot_data,
  asp = 1, ylab = "Likelihood",
  xlim = c(0, 1), ylim = c(0, 0.5)
)
```



(d) 根据 (c) 中的图，likelihood 在 prob 为 0.5 时达到最大，因此 ML 估计值为 0.5

第二章　列联表

2.1　列联表的概率结构

关于来世

```
library(cdabookdb)
data("afterlife1")
afterlife1
```

```
##          Belief
## Gender    Yes No or Undecided
##   Females 509             116
##   Males   398             104
```

```
margin.table(afterlife1, margin = 1)  # 求行和
```

```
## Gender
## Females   Males
##     625     502
```

```
margin.table(afterlife1, margin = 2)  # 求列和
```

```
## Belief
##             Yes No or Undecided
##             907             220
```

```
addmargins(afterlife1)  # 将行列求和加入列联表
```

```
##          Belief
## Gender     Yes No or Undecided  Sum
##   Females  509             116  625
##   Males    398             104  502
```

```
##   Sum      907             220 1127
```

```
prop.table(afterlife1, margin = 1)  # 求给定行的条件分布
```

```
##          Belief
## Gender       Yes No or Undecided
##   Females 0.8144          0.1856
##   Males   0.7928          0.2072
```

```
prop.table(afterlife1, margin = 2)  # 求给定列的条件分布
```

```
##          Belief
## Gender       Yes No or Undecided
##   Females 0.5612          0.5273
##   Males   0.4388          0.4727
```

2.2 2×2 表比例的比较

阿司匹林与心脏病（列联表检验）

```
library(cdabookdb)
data("aspirin")
aspirin
```

```
##          MI
## Group         Y     N
##   Placebo   189 10845
##   Aspirin   104 10933
```

```
margin.table(aspirin, 1)  # 服用安慰剂和阿司匹林的人数
```

```
## Group
## Placebo Aspirin
##   11034   11037
```

```
prop.table(aspirin, 1)  # 两个组中患心肌梗死的比例
```

```
##          MI
## Group            Y        N
##   Placebo 0.017129 0.982871
```

```
##   Aspirin 0.009423 0.990577
```

```
prop.test(aspirin)  # 对两个患病比率是否相同进行检验并求出置信区间
```

```
##
##  2-sample test for equality of proportions with
##  continuity correction
##
## data:  aspirin
## X-squared = 24, df = 1, p-value = 8e-07
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  0.004597 0.010815
## sample estimates:
##   prop 1   prop 2
## 0.017129 0.009423
```

2.3 优势比

阿司匹林与心脏病（优势比）

计算优势比可使用本文档配套包 cdabookcode 里的 oddsratio 计算，使用详情请?oddsratio

```
library(cdabookfunc)
library(cdabookdb)
data("aspirin")
oddsratio(aspirin)  # 优势比
```

```
## [1] 1.832
```

吸烟状态与心肌梗死

```
library(cdabookfunc)
library(cdabookdb)
data("smoking_mi")
oddsratio(smoking_mi, row_id = c(2, 1))  # 优势比
```

```
## [1] 3.822
```

2.4 独立性的卡方检验

性别和党派认同

```
library(cdabookfunc)
library(cdabookdb)
data("gender_party")
oddsratio(gender_party, col_id = c(1, 3))  # 优势比
```

```
## [1] 1.605
```

卡方检验可直接使用 chisq.test() 完成

```
# X2 test
x2_result <- chisq.test(gender_party)  # 独立性卡方检验
x2_result
```

```
##
##  Pearson's Chi-squared test
##
## data:  gender_party
## X-squared = 30, df = 2, p-value = 3e-07
```

G2 统计量的计算需要先得到独立性假设下的期望值，也可从 chisq.test() 的结果中得到

```
# G2
gender_party_expected <- x2_result$expected  # 获取独立性假设下的期望值
gender_party_expected
```

```
##          Party
## Gender    Democrat Independent Republican
##   Females    703.7       319.6      533.7
##   Males      542.3       246.4      411.3
```

```
Gsq <- 2 * sum(gender_party * log(gender_party / gender_party_expected))
pvalue <- 1 - pchisq(Gsq, 2)
Gsq; pvalue
```

```
## [1] 30.02
```

```
## [1] 3.034e-07
```

此外，X2 和 G2 检验也可以使用 cdabookcode 中的 independent_test_of_table() 实现

```
independent_test_of_table(gender_party, "X2")
```

```
## $method
## [1] "X2"
##
## $statistic
## [1] 30.07
##
## $df
## [1] 2
##
## $p.value
## [1] 2.954e-07
```

```
independent_test_of_table(gender_party, "G2")
```

```
## $method
## [1] "G2"
##
## $statistic
## [1] 30.02
##
## $df
## [1] 2
##
## $p.value
## [1] 3.034e-07
```

残差和标准化残差同样可以从 `chisq.test()` 的结果中得到

```
# 残差
gender_party - gender_party_expected
```

```
##          Party
## Gender    Democrat Independent Republican
##   Females   58.329       7.355    -65.683
##   Males    -58.329      -7.355     65.683
```

```
# 标准化残差
x2_result$stdres
```

```
##          Party
```

```
## Gender    Democrat Independent Republican
##   Females   4.5021      0.6995    -5.3159
##   Males    -4.5021     -0.6995     5.3159
```

2.5 有序数据的独立性检验

饮酒与婴儿畸形

M2 检验也可以使用 independent_test_of_table() 实现

```
library(cdabookfunc)
library(cdabookdb)
data("malformation")
# 对比 X2, G2, M2 的结果
# 使用 method="all" 可以同时进行 X2, G2, M2 检验
independent_test_of_table(malformation, "all", c(0, 0.5, 1.5, 4, 7), 0:1)
```

```
## Warning in chisq.test(x): Chi-squared approximation may be
## incorrect
```

```
##      method statistic df p.value
## [1,] "X2"   12.08     4  0.01675
## [2,] "G2"   6.202     4  0.1846
## [3,] "M2"   6.57      1  0.01037
```

u 和 v 的选取会影响结果

```
independent_test_of_table(malformation, "G2", 1:5, 0:1)
```

```
## $method
## [1] "G2"
##
## $statistic
## [1] 6.202
##
## $df
## [1] 4
##
## $p.value
## [1] 0.1846
```

2.6 小样本的精确推断

女士品茶

```
# 计算概率（超几何分布）
dhyper(0:4, 4, 4, 4)
```

```
## [1] 0.01429 0.22857 0.51429 0.22857 0.01429
```

费雪精确检验可使用 `fisher.test()`

```
tea_tasting <- matrix(c(3, 1, 1, 3), nrow = 2)
fisher.test(tea_tasting, alternative = "g")
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  tea_tasting
## p-value = 0.2
## alternative hypothesis: true odds ratio is greater than 1
## 95 percent confidence interval:
##  0.3136    Inf
## sample estimates:
## odds ratio
##      6.408
```

```
fisher.test(tea_tasting, alternative = "t")
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  tea_tasting
## p-value = 0.5
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##    0.2117 621.9338
## sample estimates:
## odds ratio
##      6.408
```

2.7 三项列联表的关联性

死刑判决案例

```
library(cdabookfunc)
library(cdabookdb)
data("deathpenalty1")
ftable(deathpenalty1)
```

```
##                  DeathPenalty Yes  No
## Defendant Victim
## White     White                53 414
##           Black                 0  16
## Black     White                11  37
##           Black                 4 139
```

```
# 被判死刑的比例
prop.table(deathpenalty1, c(1, 2))[, , 1]
```

```
##           Victim
## Defendant   White   Black
##     White 0.11349 0.00000
##     Black 0.22917 0.02797
```

```
# 根据被告种族，被判死刑的比例
prop.table(margin.table(deathpenalty1, margin = c(1, 3)), margin = 1)[, 1]
```

```
##   White   Black
## 0.10973 0.07853
```

```
# 受害者为白人时的优势比（条件优势比）
oddsratio(deathpenalty1[, 1, ])
```

```
## [1] 0.4306
```

```
# 不考虑受害者时的优势比（边际优势比）
oddsratio(margin.table(deathpenalty1, c(1, 3)))
```

```
## [1] 1.446
```

临床试验

```
library(cdabookfunc)
library(cdabookdb)
data("treatment1")

# 条件优势比 (clinic=1)
oddsratio(treatment1[1, ,])
```

```
## [1] 1
```

```
# 条件优势比 (clinic=1)
oddsratio(margin.table(treatment1, c(2, 3)))
```

```
## [1] 2
```

课后题

第 18 题

(a)

```
library(cdabookdb)
data("happiness1")
happiness1
```

```
##           Happiness
## Income     NotTooHappy PrettyHappy VeryHappy
##   AboveAvg          21          159       110
##   Avg               53          372       221
##   BelowAvg          94          249        83
```

The formula to calculate the estimated expected cell count is $\hat{\mu}_{11} = \frac{n_{1+}n_{+1}}{n}$

```
mu <- rowSums(happiness1)[1] * colSums(happiness1)[1] / sum(happiness1)
unname(mu)
```

```
## [1] 35.77
```

Then we get that $\hat{\mu}_{11} = 35.8$

(b) The formula to calculate df is $df = (I-1)(J-1)$

```
df <- prod(dim(happiness1) - 1)
Pv <- 1 - pchisq(73.4, df)
df; Pv
```

```
## [1] 4
```

```
## [1] 4.33e-15
```

Then we get that $df = 4$, $pvalue = 0$.

(c) These show a greater discrepancy between n_{11} and $\hat{\mu}_{11}$ (n_{33} and$\hat{\mu}_{33}$) than we would expect if the variables were truly independent. There have large negative residuals for above average income not very happy person and below average income very happy person. Thus, there were fewer people than the hypothesis of independence predicts.

(d) There have large positive residuals for above average income very happy person and below average income not very happy person. Thus, there were more people than the hypothesis of independence predicts.

第 22 题

(a)

```
library(cdabookdb)
data("psych_diag_drugs")
psych_diag_drugs
```

```
##                       Drugs
## Diagnosis                Y   N
##   Schizophrenia        105   8
##   AffectiveDisorder     12   2
##   Neurosis              18  19
##   PersonalityDisorder   47  52
##   SpecialSymptoms        0  13
```

$$\hat{\mu}_{11} = \frac{n_{1+}n_{+1}}{n}$$

$$SR = \frac{n_{ij} - \hat{\mu}_{ij}}{\sqrt{\hat{\mu}_{ij}(1 - p_{i+})(1 - p_{+j})}}$$

$$X^2 = \sum \frac{(n_{ij} - \hat{\mu}_{ij})^2}{\hat{\mu}_{ij}}$$

```
# X-squared, df and p-value
chisq.test(psych_diag_drugs)
```

```
## Warning in chisq.test(psych_diag_drugs): Chi-squared
## approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  psych_diag_drugs
## X-squared = 84, df = 4, p-value <2e-16
```

```
# standard residual
chisq.test(psych_diag_drugs)$stdres
```

```
## Warning in chisq.test(psych_diag_drugs): Chi-squared
## approximation may be incorrect
```

```
##                      Drugs
## Diagnosis                  Y      N
##   Schizophrenia        7.875 -7.875
##   AffectiveDisorder    1.602 -1.602
##   Neurosis            -2.385  2.385
##   PersonalityDisorder -4.842  4.842
##   SpecialSymptoms     -5.139  5.139
```

We could obtain $X^2 = 84.188, df = 4$,and P-value almost equals to 0.Thus, we should reject null hypothesis, which means psychiatric diagnosis and whether patients have drugs are not independent. Positive standardized residuals means there are more people than expected and negative standardized residuals means there are less people than expected. If the absolute value of standardized residuals less than 2, then we do not have strong evidence to reject the null hypothesis.

(c)

i). the first two rows

```
psy12 <- psych_diag_drugs[1:2,]
chisq.test(psy12)
```

```
## Warning in chisq.test(psy12): Chi-squared approximation may
## be incorrect
```

```
##
##  Pearson's Chi-squared test with Yates' continuity
```

```
##  correction
##
## data:  psy12
## X-squared = 0.17, df = 1, p-value = 0.7
```

Then we get $X^2 = 0.175$, df=1, and P-value = 0.6757,then we can not reject null hypothesis.

ii). the third and fourth rows

```
psy34 <- psych_diag_drugs[3:4,]
chisq.test(psy34)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity
##  correction
##
## data:  psy34
## X-squared = 1.4e-30, df = 1, p-value = 1
```

Then we get X^2 almost equals to 0 , df=1, and P-value = 1,then we can not reject null hypothesis.

iii). the last row to the first and second rows combined and the third and fourth rows combined

```
psy0 <- rbind(colSums(psy12), colSums(psy34), psych_diag_drugs[5, ])
chisq.test(psy0)
```

```
## Warning in chisq.test(psy0): Chi-squared approximation may
## be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  psy0
## X-squared = 84, df = 2, p-value <2e-16
```

Then we get $X^2 = 83.884$, df=2, and P-value almost equals to 0 ,then we can reject null hypothesis, psychiatric diagnosis and whether patients have drugs are not independent.

第 33 题

(a)

```
library(cdabookfunc)
library(cdabookdb)
```

```
data("deathpenalty2")
ftable(deathpenalty2)
```

```
##                  DeathPenalty Yes  No
## Defendant Victim
## White     White                19 132
##           Black                 0   9
## Black     White                11  52
##           Black                 6  97
```

(b)

```
# When victim is white
deathpenalty2[, 1, ]
```

```
##          DeathPenalty
## Defendant Yes  No
##     White  19 132
##     Black  11  52
```

```
oddsratio(deathpenalty2[, 1, ], 0.5)
```

```
## [1] 0.6719
```

```
# When victim is black
deathpenalty2[, 2, ]
```

```
##          DeathPenalty
## Defendant Yes No
##     White   0  9
##     Black   6 97
```

```
oddsratio(deathpenalty2[, 2, ], 0.5)
```

```
## [1] 0.7895
```

Controlling for victims' race, the percentage of "yes" death penalty verdicts was higher for black defendants than for white defendants.

(c)

```
# Ignorevictims' race
margin.table(deathpenalty2, 2)
```

```
## Victim
## White Black
##   214   112
```

```
oddsratio(margin.table(deathpenalty2, c(1, 3)))
```

```
## [1] 1.181
```

Ignoring for victims' race, the percentage of "yes" death penalty verdicts was higher for black defendants than for white defendants.

Then these data exhibit Simpson's paradox.

第三章　广义线性模型

3.1　广义线性模型的构成部分

3.2　二分数据的广义线性模型

打鼾与心脏病

```
library(cdabookdb)
data("snoring_heartdisease")
snoring_heartdisease
```

```
##                       Heartdisease
## Snoring                 Yes    No
##   Never                  24 1355
##   Occasional             35  603
##   Nearly every night     21  192
##   Every night            30  224
```

对打鼾数据（二分数据）拟合模型时，可以设定 `family=binomial()`，其中 `binomial()` 的 `link` 参数为 `identity`、`logit`、`probit` 时，分别表示拟合线性概率模型、logistics 模型和 probit 模型

以下使用打鼾频率得分 0, 2, 4, 5 拟合三个模型，并获取对应的预测概率

```
scores <- c(0, 2, 4, 5)

snoring_linear <- glm(
  snoring_heartdisease ~ scores, family = binomial(link = "identity")
)
snoring_logistics <- glm(
  snoring_heartdisease ~ scores, family = binomial(link = "logit")
)
```

```r
snoring_probit <- glm(
  snoring_heartdisease ~ scores, family = binomial(link = "probit")
)

model_list <- list(snoring_linear, snoring_logistics, snoring_probit)
```

```r
# 模型系数
estimated_coef <- sapply(model_list, coef)
colnames(estimated_coef) <- c("linear", "logit", "probit")
round(estimated_coef, digits = 3)
```

```
##             linear  logit probit
## (Intercept)  0.017 -3.866 -2.061
## scores       0.020  0.397  0.188
```

得到的三个模型为

$$\hat{\pi}(x) = \hat{\alpha} + \hat{\beta}x = 0.017 + 0.020x$$
$$\text{logit}(\hat{\pi}(x)) = \hat{\alpha} + \hat{\beta}x = -3.866 + 0.397x$$
$$\text{probit}(\hat{\pi}(x)) = \hat{\alpha} + \hat{\beta}x = -2.061 + 0.188x$$

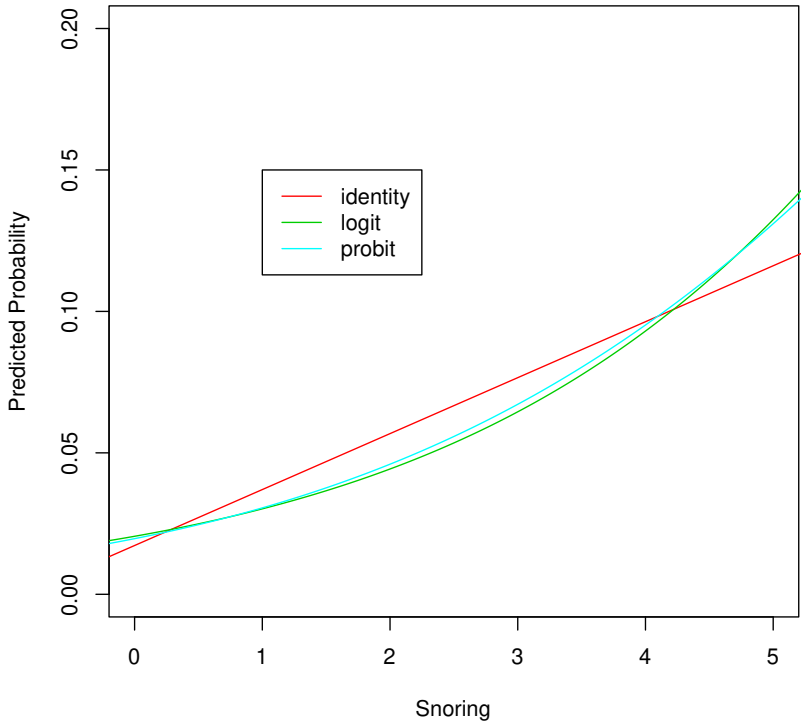
```r
# 模型预测概率
pred_prob <- sapply(model_list, predict, type = "response")
colnames(pred_prob) <- c("linear", "logit", "probit")
round(pred_prob, digits = 3)
```

```
##                    linear logit probit
## Never               0.017 0.021  0.020
## Occasional          0.057 0.044  0.046
## Nearly every night  0.096 0.093  0.095
## Every night         0.116 0.132  0.131
```

作图，在一张图中画出三个模型的图像

```r
snoring_new <- data.frame(scores=seq(-1, 6, 0.01))
plot(
  NULL,
  xlim = c(0, 5), ylim = c(0, 0.2),
  xlab = "Snoring", ylab = "Predicted Probability"
)

line_col <- c(identity = 2, logit = 3, probit = 5)
sapply(model_list, function(m) {
```

```
    pred_result <- predict(m, snoring_new, type = "response")
    lines(
      snoring_new$scores, pred_result, type = "l",
      lty = 1, col = line_col[m$family$link]
    )
  }
)
legend(1, 0.15, names(line_col), col = line_col, lty = 1)
```



3.3 计数数据的广义线性模型

母鲎及其追随者（泊松 GLM）

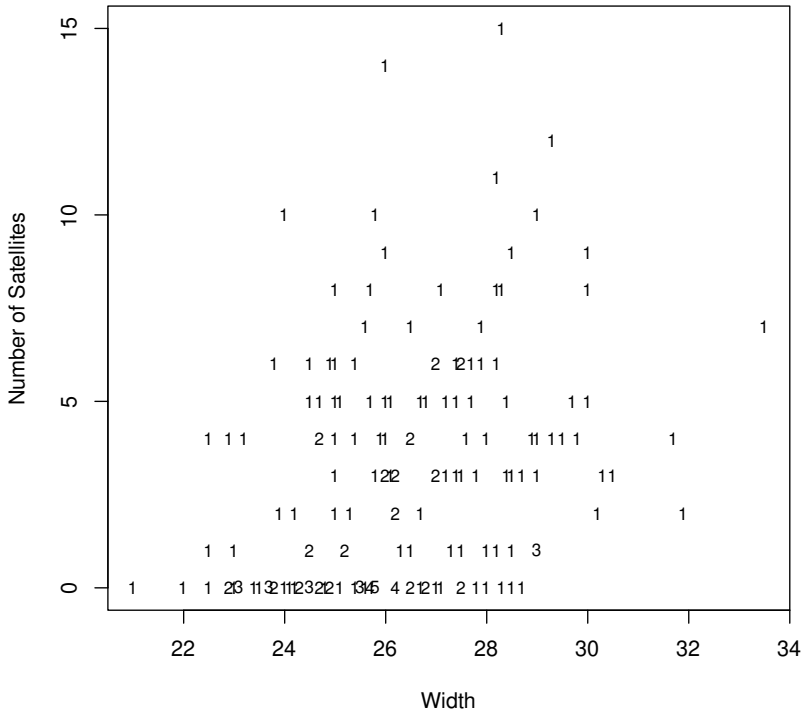
```
library(cdabookdb)
data("horseshoecrabs") # 母鲎及其追随者的数据集
head(horseshoecrabs)
```

```
##   Color Spine Width Weight Satellites
## 1     2     3  28.3   3.05          8
## 2     3     3  22.5   1.55          0
## 3     1     1  26.0   2.30          9
## 4     3     3  24.8   2.10          0
```

```
## 5      3      3  26.0    2.60           4
## 6      2      3  23.8    2.10           0
```

首先可以作出响应计数对宽度的图像，图中数字为对应点的观测数。

```
library(dplyr)
horseshoecrabs %>%
  group_by(Satellites, Width) %>%
  summarise(n = n()) %>%
  plot(
    Satellites ~ Width, data = .,
    pch = as.character(n),  # 点类型设为数字
    xlab = "Width", ylab = "Number of Satellites",  # 横纵坐标标签
    cex = 0.8  # 字体大小
  )
```



在使用该数据建模时，泊松对数线性模型可以通过在 `glm` 中设置 `family=poisson`，在 R 里泊松回归的默认联系 (`link`) 为对数，因此在这里不需要修改 `link`

```
m1 <- glm(Satellites ~ Width, family = poisson(), data = horseshoecrabs)
summary(m1)
```

```
##
## Call:
## glm(formula = Satellites ~ Width, family = poisson(), data = horseshoecrabs)
##
## Deviance Residuals:
```

```
##    Min      1Q  Median      3Q     Max
## -2.853  -1.988  -0.493   1.097   4.922
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -3.305      0.542   -6.09  1.1e-09 ***
## Width          0.164      0.020    8.22  < 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: 632.79  on 172  degrees of freedom
## Residual deviance: 567.88  on 171  degrees of freedom
## AIC: 927.2
##
## Number of Fisher Scoring iterations: 6
```

可以得出拟合的对数线性模型为

$$\log \hat{\mu} = \hat{\alpha} + \hat{\beta}x = -3.305 + 0.164x$$

而如果要拟合恒等联系的泊松模型，需要设置 `poisson(link="identity")`。另外在此案例中，直接跑回归会出现如下错误：

```
Error: no valid set of coefficients has been found: please supply starting values
In addition: Warning message:
In log(y/mu) : NaNs produced
```

即需要指定寻找最优值过程的初值，否则有可能找不到解，这里可使用对数联系的泊松模型的系数作为初值

```
m2 <- glm(
  Satellites ~ Width,
  family = poisson(link = "identity"),  # 恒等联系的泊松模型
  data = horseshoecrabs,
  start = coef(m1)  # 使用 m1 的系数作为初值
)
summary(m2)
```

```
##
## Call:
## glm(formula = Satellites ~ Width, family = poisson(link = "identity"),
##     data = horseshoecrabs, start = coef(m1))
##
```

```
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.911  -1.960  -0.541   1.041   4.799
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -11.5255     0.6777   -17.0   <2e-16 ***
## Width         0.5492     0.0297    18.5   <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: 632.79  on 172  degrees of freedom
## Residual deviance: 557.71  on 171  degrees of freedom
## AIC: 917
##
## Number of Fisher Scoring iterations: 22
```

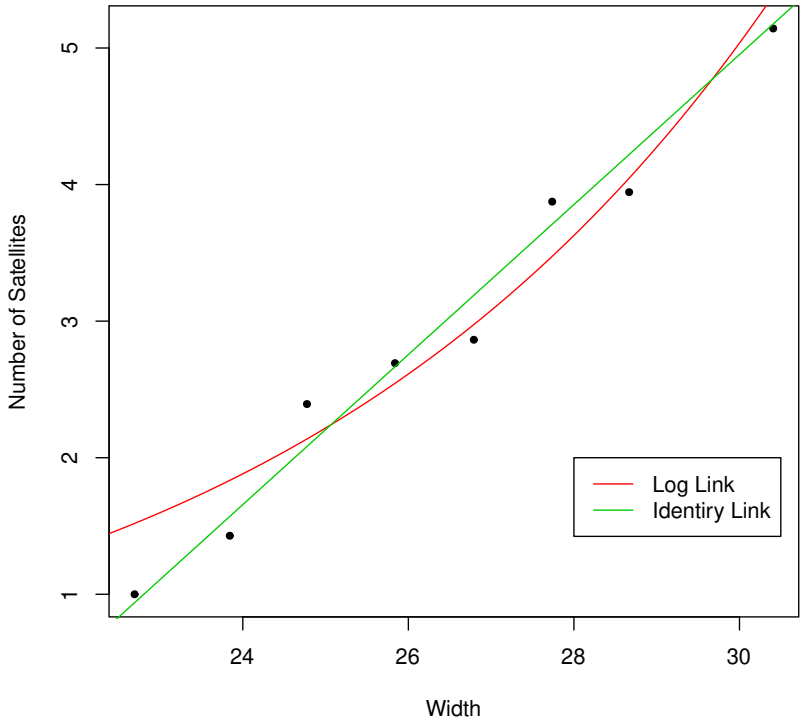
可得拟合出的模型为

$$\hat{\mu} = \hat{\alpha} + \hat{\beta}x = -11.525 + 0.549x$$

最后可以查看恒等联系和对数联系的模型估计的差别，即教材中的图 3.6。

```r
# 需要先按宽度分组，再求各组的平均宽度和平均追随者只数
mean_satellite_width <- horseshoecrabs %>%
  mutate(width_group = cut(Width, c(0, 23.25 + 0:6, Inf), dig.lab = 4)) %>% # 分区间
  group_by(width_group) %>% # 声明按 width_group 进行分组
  summarise(
    mean_width = mean(Width),  # 平均宽度
    mean_satellite = mean(Satellites)  # 平均追随者只数
  )

plot(
  mean_satellite ~ mean_width,
  data = mean_satellite_width, # 选定数据集
  pch = 20, # 点类型为实心圆点
  xlab = "Width", ylab = "Number of Satellites" # 横纵坐标标签
)

x <- seq(22, 32, 0.1)
y_m1 <- predict(m1, data.frame(Width = x), type = "response")
```

```
y_m2 <- predict(m2, data.frame(Width = x))
lines(x, y_m1, type = "l", col = 2)
lines(x, y_m2, type = "l", col = 3)
legend(28, 2, c("Log Link", "Identiry Link"), col = c(2, 3), lty = 1)
```



母蛏及其追随者（负二项 GLM）

负二项 GLM 与泊松 GLM 类似，但设定 `glm` 函数中的 `family` 参数时，R 中并不自带负二项分布的 `family`，需要使用 `MASS` 包中的 `negative.binomial()`。该函数的默认 `link` 为对数，但需要额外指定一个参数 `theta`，该参数的意义为教材 3.3.4 节中 `D` 的倒数。

需要指定 θ 的原因（应该）是 `glm()` 函数不带有寻找最优 θ 的过程，这里使用教材中最后得出的 $\widehat{D} = 1.1$ 的倒数为 θ 的值。

```
library(cdabookdb)
library(MASS)
m1 <- glm(
  Satellites ~ Width,
  family = negative.binomial(theta = 1 / 1.1),
  data = horseshoecrabs
)
summary(m1)
```

```
##
## Call:
```

```
## glm(formula = Satellites ~ Width, family = negative.binomial(theta = 1/1.1),
##     data = horseshoecrabs)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.782  -1.412  -0.251   0.478   2.022
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -4.0514     1.0777   -3.76  0.00023 ***
## Width         0.1920     0.0405    4.74  4.5e-06 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(0.9091) family taken to be 0.8495)
##
##     Null deviance: 213.63  on 172  degrees of freedom
## Residual deviance: 196.33  on 171  degrees of freedom
## AIC: 755.3
##
## Number of Fisher Scoring iterations: 5
```

另一个更优的拟合负二项 GLM 的办法是使用 `MASS` 包中的 `glm.nb()`，该函数带有寻找最优 θ 的过程，不需要指定 θ 参数。

```
m2 <- glm.nb(Satellites ~ Width, data = horseshoecrabs)
summary(m2)
```

```
##
## Call:
## glm.nb(formula = Satellites ~ Width, data = horseshoecrabs, init.theta = 0.90456808,
##     link = log)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.780  -1.411  -0.250   0.477   2.018
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -4.0525     1.1714   -3.46  0.00054 ***
## Width         0.1921     0.0441    4.36  1.3e-05 ***
## ---
```

```
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(0.9046) family taken to be 1)
##
##     Null deviance: 213.05  on 172  degrees of freedom
## Residual deviance: 195.81  on 171  degrees of freedom
## AIC: 757.3
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  0.905
##           Std. Err.:  0.161
##
##  2 x log-likelihood:  -751.291
```

英国的火车事故

```
library(cdabookdb)
library(MASS)
data("traincollisions")
head(traincollisions)
```

```
##   Year  KM Train TrRd
## 1 2003 518     0    3
## 2 2002 516     1    3
## 3 2001 508     0    4
## 4 2000 503     1    3
## 5 1999 505     1    2
## 6 1998 487     0    4
```

根据 3.5 节，使用泊松 glm 拟合比率数据时，模型为

$$\log(\mu/t) = \log(\mu) - \log(t) = \alpha + \beta x$$

由于泊松 glm 的 y 需要是正整数，因此可以把对数比率 $(\log(\mu/t))$ 化为两个对数相减 $(\log(\mu) - \log(t))$，再把 $\log(t)$ 作为 `offset`

对于 `glm` 函数，有一个参数 `offset` 可以直接设置

```
traincollisions$year0 <- traincollisions$Year - 1975
m_poisson <- glm(
```

```
  TrRd ~ year0,
  data = traincollisions, family = poisson(),
  offset = log(traincollisions$KM)
)
summary(m_poisson)
```

```
##
## Call:
## glm(formula = TrRd ~ year0, family = poisson(), data = traincollisions,
##     offset = log(traincollisions$KM))
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.058  -0.783  -0.083   0.377   3.387
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -4.2114     0.1589  -26.50   <2e-16 ***
## year0        -0.0329     0.0108   -3.06   0.0022 **
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 47.376  on 28  degrees of freedom
## Residual deviance: 37.853  on 27  degrees of freedom
## AIC: 133.5
##
## Number of Fisher Scoring iterations: 5
```

得到的模型为

$$\log(\hat{\mu}) - \log(t) = -4.2114 - 0.0329x$$

而负二项 glm 使用的 `glm.nb()` 函数没有 `offset` 参数，因此可利用 `offset()` 函数将其纳入 `formula` 中。

```
m_nb <- glm.nb(
  TrRd ~ year0 + offset(log(KM)),
  data = traincollisions
)
summary(m_nb)
```

```
##
## Call:
## glm.nb(formula = TrRd ~ year0 + offset(log(KM)), data = traincollisions,
##     init.theta = 10.11828724, link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7237  -0.6546  -0.0587   0.3298   2.6407
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -4.2000     0.1958  -21.45   <2e-16 ***
## year0        -0.0337     0.0129   -2.61   0.0089 **
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(10.12) family taken to be 1)
##
##     Null deviance: 32.045  on 28  degrees of freedom
## Residual deviance: 25.264  on 27  degrees of freedom
## AIC: 132.7
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  10.12
##           Std. Err.:  8.00
##
##  2 x log-likelihood:  -126.69
```

得到模型为

$$\log(\hat{\mu}) - \log(t) = -4.2000 - 0.0337x$$

3.4 统计推断和模型检验

打鼾与心脏病

```
library(cdabookdb)
data("snoring_heartdisease")
scores <- c(0, 2, 4, 5)
```

```
snoring_linear <- glm(
  snoring_heartdisease ~ scores,
  family = binomial(link = "identity")
)
summary(snoring_linear)
```

```
##
## Call:
## glm(formula = snoring_heartdisease ~ scores, family = binomial(link = "identity"))
##
## Deviance Residuals:
##              Never          Occasional  Nearly every night
##             0.0448             -0.2132              0.1101
##        Every night
##             0.0980
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.01725    0.00345    5.00  5.8e-07 ***
## scores       0.01978    0.00280    7.05  1.8e-12 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 65.904481  on 3  degrees of freedom
## Residual deviance:  0.069191  on 2  degrees of freedom
## AIC: 24.32
##
## Number of Fisher Scoring iterations: 3
```

```
anova(snoring_linear, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: identity
##
## Response: snoring_heartdisease
##
## Terms added sequentially (first to last)
##
```

```
##
##        Df Deviance Resid. Df Resid. Dev Pr(>Chi)
## NULL                       3       65.9
## scores  1     65.8         2        0.1  4.9e-16 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

可使用 `confint()` 来得到 likelihood ratio 置信区间，但需要先得到载入 `MASS` 包

```
library(MASS)
confint(snoring_linear)
```

```
##                2.5 %  97.5 %
## (Intercept) 0.01133 0.02483
## scores      0.01452 0.02551
```

3.5 广义线性模型的拟合

课后题

第 3 题

(a) 列出预测等式:

$$y = 0.0025 + 0.0011 * alcohol$$

斜率 0.0011 表示，饮酒水平每增加 1，婴儿畸形的概率上升 0.0011

(b)

$$y_1 = 0.0025 + 0.0011 \times 0 = 0.0025$$
$$y_2 = 0.0025 + 0.0011 \times 7 = 0.0102$$

饮酒量为 0 时，畸形的概率为 0.0025；饮酒量为 7.0 时，畸形的概率为 0.0102,；相对风险 0.245

第 4 题

(a) 敏感

首先拟合第三题中的模型

注意拟合线性概率模型时使用的矩阵需要第一列表示 “成功” 的数量，所以这里数据的两列需要先互换

```
library(cdabookdb)
data("malformation")
alcohol_score <- c(0, 0.5, 1.5, 4, 7)  # 饮酒量得分
# 更换顺序两列的顺序
# 拟合线性概率模型时使用的矩阵需要第一列表示“成功”的数量
malformation0 <- malformation[, 2:1]
malformation0
```

```
##        Malformation
## Alcohol Present Absent
##     0        48  17066
##     <1       38  14464
##     1-2       5    788
##     3-5       1    126
##     >=6       1     37
```

```
m_problem3 <- glm(
  malformation0 ~ alcohol_score, family = binomial("identity")
)
summary(m_problem3)
```

```
##
## Call:
## glm(formula = malformation0 ~ alcohol_score, family = binomial("identity"))
##
## Deviance Residuals:
##      0      <1     1-2     3-5     >=6
##  0.656  -1.049   0.863   0.130   0.828
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.002548   0.000352    7.23  4.8e-13 ***
## alcohol_score 0.001087   0.000832    1.31     0.19
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6.2020  on 4  degrees of freedom
## Residual deviance: 2.9795  on 3  degrees of freedom
## AIC: 25.61
```

```
##
## Number of Fisher Scoring iterations: 10
```

删除一个观测重新拟合

```
malformation1 <- malformation0
malformation1[5, 1] <- 0
malformation1
```

```
##        Malformation
## Alcohol Present Absent
##     0        48  17066
##     <1       38  14464
##     1-2       5    788
##     3-5       1    126
##     >=6       0     37
```

```
m_problem4a <- glm(
  malformation1 ~ alcohol_score, family = binomial("identity")
)
summary(m_problem4a)
```

```
##
## Call:
## glm(formula = malformation1 ~ alcohol_score, family = binomial("identity"))
##
## Deviance Residuals:
##      0      <1     1-2     3-5     >=6
##  0.430  -0.791   1.127   0.369  -0.738
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.002635   0.000352    7.48  7.5e-14 ***
## alcohol_score 0.000672   0.000785    0.86     0.39
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3.7225  on 4  degrees of freedom
## Residual deviance: 2.7609  on 3  degrees of freedom
## AIC: 23.41
```

```
##
## Number of Fisher Scoring iterations: 6
```

```
cbind(
  `model in problem 3`=coef(m_problem3),
  `model in problem 4(a)`=coef(m_problem4a)
)
```

```
##               model in problem 3 model in problem 4(a)
## (Intercept)             0.002548             0.0026346
## alcohol_score           0.001087             0.0006716
```

饮酒量的参数从 0.00109 下降到了 0.00067，说明模型对那个观测敏感。

(b) 敏感

更换饮酒量得分，重新拟合

```
alcohol_score_4b <- 0:4
m_problem4b <- glm(
  malformation0 ~ alcohol_score_4b, family = binomial("identity")
)
summary(m_problem4b)
```

```
##
## Call:
## glm(formula = malformation0 ~ alcohol_score_4b, family = binomial("identity"))
##
## Deviance Residuals:
##      0      <1     1-2     3-5     >=6
##  0.525  -1.072   1.145   0.588   1.360
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)      0.002598   0.000380    6.84  7.8e-12 ***
## alcohol_score_4b 0.000504   0.000528    0.96     0.34
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6.2020  on 4  degrees of freedom
## Residual deviance: 4.9336  on 3  degrees of freedom
```

```
## AIC: 27.56
##
## Number of Fisher Scoring iterations: 9
```

```
cbind(
  `model in problem 3`=coef(m_problem3),
  `model in problem 4(b)`=coef(m_problem4b)
)
```

```
##               model in problem 3 model in problem 4(b)
## (Intercept)             0.002548             0.0025977
## alcohol_score           0.001087             0.0005044
```

```
alcohol_score_4b / alcohol_score
```

```
## [1]    NaN 2.0000 1.3333 0.7500 0.5714
```

饮酒量的参数大概只是原来的一半，而饮酒量得分只有当饮酒量小于 1 时才是原来的两倍。

```
cbind(
  `pred-prob in 3` = predict(m_problem3)[c(1, 5)],
  `pred-prob in 4(b)` = predict(m_problem4b)[c(1, 5)]
)
```

```
##     pred-prob in 3 pred-prob in 4(b)
## 0         0.002548          0.002598
## >=6       0.010158          0.004615
```

预测出来的饮酒量也有很大差异（在最大饮酒量时）。

(c) 拟合 logit 和 probit 模型

```
m_problem4c_logit <- glm(
  malformation0 ~ alcohol_score,
  family = binomial("logit")
)
m_problem4c_probit <- glm(
  malformation0 ~ alcohol_score,
  family = binomial("probit")
)


cbind(
  `linear` = coef(m_problem3),
```

```
  `logit` = coef(m_problem4c_logit),
  `probit` = coef(m_problem4c_probit)
)
```

```
##                 linear   logit  probit
## (Intercept)   0.002548 -5.9605 -2.7996
## alcohol_score 0.001087  0.3166  0.1098
```

从而三个模型为

$$\hat{\pi}(x) = \hat{\alpha} + \hat{\beta}x = 0.00255 + 0.00109x$$
$$\text{logit}(\hat{\pi}(x)) = \hat{\alpha} + \hat{\beta}x = -5.96046 + 0.31656x$$
$$\text{probit}(\hat{\pi}(x)) = \hat{\alpha} + \hat{\beta}x = -2.79961 + 0.10979x$$

三个模型饮酒量的系数都为正，表明随着饮酒程度的增加，婴儿畸形概率增大

第 7 题

在做题前需要先按题目要求，构造出 Y 变量，当蛏的追随者大于 0 时，Y=1，否则 Y=0

```
library(cdabookdb)
data(horseshoecrabs)
# psat 即为题目中的 Y
horseshoecrabs$psat <- as.integer(horseshoecrabs$Satellites > 0)
```

(a) OLS 估计可直接使用 lm() 函数

```
m1 <- lm(psat ~ Weight, data = horseshoecrabs)
summary(m1)
```

```
##
## Call:
## lm(formula = psat ~ Weight, data = horseshoecrabs)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -0.888 -0.468  0.161  0.370  0.669
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.1449     0.1472   -0.98     0.33
## Weight        0.3227     0.0588    5.49  1.4e-07 ***
```

```
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.445 on 171 degrees of freedom
## Multiple R-squared:  0.15,   Adjusted R-squared:  0.145
## F-statistic: 30.2 on 1 and 171 DF,  p-value: 1.42e-07
```

得到模型为

$$Y = -0.155 + 0.323weight$$

从而体重每增加 1kg，有追随者的概率就增加 0.323。

使用该模型预测当重量为 5.2kg 时有追随者的概率

```
predict(m1, newdata = data.frame(Weight = 5.2))
```

```
##     1
## 1.533
```

概率大于 1，这说明这个线性模型并不好。

(b) 尝试使用 ML 方法估计线性概率模型，会像这样报错

```
m2 <- glm(
  psat ~ Weight, data = horseshoecrabs, family = binomial(link = "identity")
)
```

```
## Error: no valid set of coefficients has been found: please supply starting values
```

报错原因是拟合的概率落到了 $(0, 1)$ 之外

(c)

```
# 拟合 logistic 模型
m3 <- glm(psat ~ Weight, data = horseshoecrabs, family = binomial())
summary(m3)
```

```
##
## Call:
## glm(formula = psat ~ Weight, family = binomial(), data = horseshoecrabs)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.111  -1.075   0.543   0.912   1.629
##
```

```
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -3.695      0.880   -4.20  2.7e-05 ***
## Weight         1.815      0.377    4.82  1.4e-06 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 225.76  on 172  degrees of freedom
## Residual deviance: 195.74  on 171  degrees of freedom
## AIC: 199.7
##
## Number of Fisher Scoring iterations: 4
```

```
# weight 为 5.2kg 时的 logit 值
predict(m3, newdata = data.frame(Weight = 5.2))
```

```
##     1
## 5.744
```

```
log(0.9968 / (1 - 0.9968))
```

```
## [1] 5.741
```

第 20 题

(a)

```
library(cdabookdb)
data("smoking_cd")
ftable(smoking_cd, row.vars = "Age", col.vars = c("Item", "Smoking"))
```

```
##       Item    Person-Years          Coronary Deaths
##       Smoking   Nonsmokers Smokers      Nonsmokers Smokers
## Age
## 35-44              18793   52407               2      32
## 45-54              10673   43248              12     104
## 55-64               5710   28612              28     206
## 65-74               2585   12663              28     186
## 75-84               1462    5317              31     102
```

计算死亡率，以及吸烟和不吸烟的死亡率比例

```
death_rate <- smoking_cd[, , 2] / smoking_cd[, , 1] * 1000
death_rate[, 2] / death_rate[, 1]
```

```
##  35-44  45-54  55-64  65-74  75-84
## 5.7376 2.1388 1.4682 1.3561 0.9047
```

可以看出，不管是不是吸烟者，冠心病死亡率都随着年龄而增长。同时，吸烟的影响随着年龄的增长而减小。

(b) 主效应模型假设了吸烟的影响不取决于年龄。这是不合理的，从 (a) 中可以明显看出吸烟的影响随着年龄而变化。

(c) 从 (a) 中可以看出吸烟的影响随着年龄增长而递减，从而我们可以给年龄赋予适当的得分来使其成为定量变量。基于此，我们可以考虑吸烟和年龄的定量交互。

而模型为

$$\log\left(\frac{deathnum}{personnum}\right) = \alpha + \beta_1 \text{age} + \beta_2 \text{smoking} + \beta_3 \text{age} \times \text{smoking}$$

其中年龄为定量变量。对于吸烟者，$\text{smoking} = 1$，则

$$\log\left(\frac{deathnum}{personnum}\right) = (\alpha + \beta_2) + (\beta_1 + \beta_3)\text{age}$$

对于非吸烟者，$\text{smoking} = 0$，则

$$\log\left(\frac{deathnum}{personnum}\right) = \alpha + \beta_1 \text{age}$$

这两个都是线性模型

(d) 拟合模型前需要先变换数据结构

```
library(tidyr)
smoking_cd_df <- spread(as.data.frame(smoking_cd), Item, Freq)
```

首先拟合 (b) 中的主效应模型

```
m1 <- glm(
  `Coronary Deaths` ~ Age + Smoking,
  offset = log(`Person-Years`),
  data = smoking_cd_df,
  family = poisson()
)
summary(m1)
```

```
##
## Call:
## glm(formula = `Coronary Deaths` ~ Age + Smoking, family = poisson(),
##     data = smoking_cd_df, offset = log(`Person-Years`))
##
## Deviance Residuals:
##       1        2        3        4        5        6
## -2.1800   0.9018  -1.3080   0.5104  -0.1379   0.0513
##       7        8        9       10
##  0.2289  -0.0873   1.9191  -0.9124
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -7.919      0.192  -41.30  < 2e-16 ***
## Age45-54          1.484      0.195    7.61  2.8e-14 ***
## Age55-64          2.628      0.184   14.30  < 2e-16 ***
## Age65-74          3.351      0.185   18.13  < 2e-16 ***
## Age75-84          3.700      0.192   19.25  < 2e-16 ***
## SmokingSmokers    0.355      0.107    3.30  0.00096 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 935.091  on 9  degrees of freedom
## Residual deviance:  12.134  on 4  degrees of freedom
## AIC: 79.2
##
## Number of Fisher Scoring iterations: 4
```

接着给 5 个年龄组分别赋予得分 1, 2, 3, 4, 5 并拟合 (c) 中的模型

```
smoking_cd_df$age_score <- as.numeric(smoking_cd_df$Age)
m2 <- glm(
  `Coronary Deaths` ~ age_score * Smoking,
  offset = log(`Person-Years`),
  data = smoking_cd_df,
  family = poisson()
)
summary(m2)
```

```
##
```

```
## Call:
## glm(formula = `Coronary Deaths` ~ age_score * Smoking, family = poisson(),
##     data = smoking_cd_df, offset = log(`Person-Years`))
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -3.878  -2.122  -0.248   1.718   3.527
##
## Coefficients:
##                           Estimate Std. Error z value
## (Intercept)               -8.8672     0.3057  -29.01
## age_score                  1.0468     0.0774   13.52
## SmokingSmokers             1.2837     0.3258    3.94
## age_score:SmokingSmokers  -0.2490     0.0836   -2.98
##                           Pr(>|z|)
## (Intercept)               < 2e-16 ***
## age_score                 < 2e-16 ***
## SmokingSmokers            8.2e-05 ***
## age_score:SmokingSmokers   0.0029 **
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 935.091  on 9  degrees of freedom
## Residual deviance:  59.895  on 6  degrees of freedom
## AIC: 123
##
## Number of Fisher Scoring iterations: 4
```

前一个模型有更小的 residual deviance 和 AIC，看起来似乎前一个模型更好。

出现这种情况的原因可能是吸烟的影响不是随着年龄而线性变化的。

第四章　logistic 回归

4.1　logistic 回归模型的解释

母蟹及其追随者（logistic 回归）

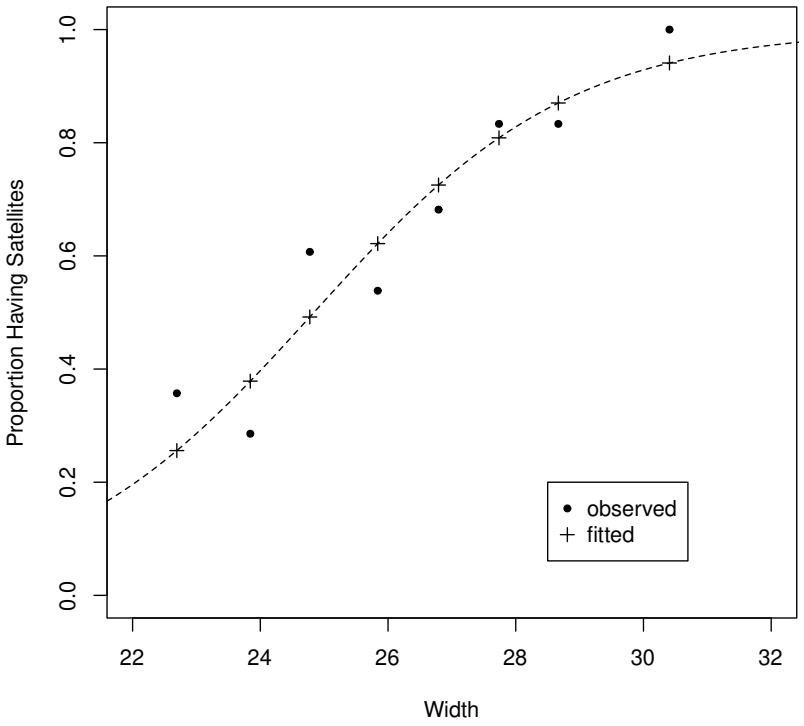
```
library(cdabookdb)
data("horseshoecrabs")
horseshoecrabs$psat <- as.integer(horseshoecrabs$Satellites > 0)

m1 <- glm(psat ~ Width, data = horseshoecrabs, family = binomial())
summary(m1)
```

```
##
## Call:
## glm(formula = psat ~ Width, family = binomial(), data = horseshoecrabs)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.028  -1.046   0.548   0.907   1.694
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -12.351      2.629   -4.70  2.6e-06 ***
## Width          0.497      0.102    4.89  1.0e-06 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 225.76  on 172  degrees of freedom
## Residual deviance: 194.45  on 171  degrees of freedom
```

```
## AIC: 198.5
##
## Number of Fisher Scoring iterations: 4
```

该模型的详细解释可从教材中得到，以下是教材中评价模型拟合情况的图（图 4.3）的一种作图方法。

```
library(dplyr)
# 需要先按宽度分组，再求各组的平均宽度和平均追随者只数
mean_width_vs_prop <- horseshoecrabs %>%
  mutate(width_group = cut(Width, c(0, 23.25 + 0:6, Inf), dig.lab = 4)) %>%
  group_by(width_group) %>% # 声明按 width_group 进行分组
  summarise(
    prop = mean(psat),  # 各分组具有追随者的比例
    mean_width = mean(Width)  # 平均宽度
  )

prop <- mean_width_vs_prop$prop  # 各个分组下具有追随者的比例
mean_width <- mean_width_vs_prop$mean_width  # 各个分组的平均宽度

# 计算各个分组的平均宽度下的预测概率
pred_prop <- predict(
  m1, data.frame(Width = mean_width), type = "response"
)

# 绘制拟合曲线的数据
width_seq <- seq(21, 33, 0.1)
pred_prop_seq <- predict(
  m1, data.frame(Width = width_seq), type = "response"
)


plot(
  prop ~ mean_width, pch = 20,  # 点类型为实心圆点
  xlim = c(22, 32), ylim = c(0, 1),  # 横纵坐标范围
  xlab = "Width", ylab = "Proportion Having Satellites"  # 横纵坐标标签
)

points(mean_width, pred_prop, pch = 3)  # 点类型为加号
points(width_seq, pred_prop_seq, type = "l", lty = 2)  # 类型为线，线类型为虚线
legend(28.5, 0.2, c("observed", "fitted"), pch = c(20, 3))  # 图例
```



4.2 logistic 回归的推断

4.3 属性预测变量的 logistic 回归

AZT 和 AIDS

```
library(cdabookdb)
data("AZT")
AZT0 <- as.data.frame(AZT)
# 构造因变量
AZT0$y <- AZT0$Symptoms == "Yes"
# 拟合模型
AZT.glm <- glm(
  y ~ (AZTUse == "Yes") + (Race == "White"),
  data = AZT0,
  weights = Freq,
  family = binomial()
)
summary(AZT.glm)
```

```
##
## Call:
```

```
## glm(formula = y ~ (AZTUse == "Yes") + (Race == "White"), family = binomial(),
##     data = AZT0, weights = Freq)
##
## Deviance Residuals:
##     1      2      3      4      5      6      7      8
##  7.29   6.54   9.21   5.73  -5.49  -4.00  -7.07  -5.03
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -1.0736     0.2629   -4.08  4.4e-05
## AZTUse == "Yes"TRUE  -0.7195     0.2790   -2.58   0.0099
## Race == "White"TRUE   0.0555     0.2886    0.19   0.8475
##
## (Intercept)         ***
## AZTUse == "Yes"TRUE **
## Race == "White"TRUE
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 342.12  on 7  degrees of freedom
## Residual deviance: 335.15  on 5  degrees of freedom
## AIC: 341.2
##
## Number of Fisher Scoring iterations: 5
```

```
# LR 检验
anova(AZT.glm, test="LRT")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: y
##
## Terms added sequentially (first to last)
##
##
##                 Df Deviance Resid. Df Resid. Dev Pr(>Chi)
## NULL                                7        342
## AZTUse == "Yes"  1     6.93         6        335   0.0085
```

```
## Race == "White"  1     0.04         5        335   0.8473
##
## NULL
## AZTUse == "Yes" **
## Race == "White"
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

4.4 多元 logistic 回归

母蛏及其追随者（多元 logistic）

在从 cdabookcode 包中将数据引入之后，由于 Color 列为数值类型，需要先转换为因子类型。此外，在回归中使用因子型变量时，R 会将因子水平的第一个作为基准类型，以下示例中为了与教材结果一致将使用颜色 4 作为基准类型。

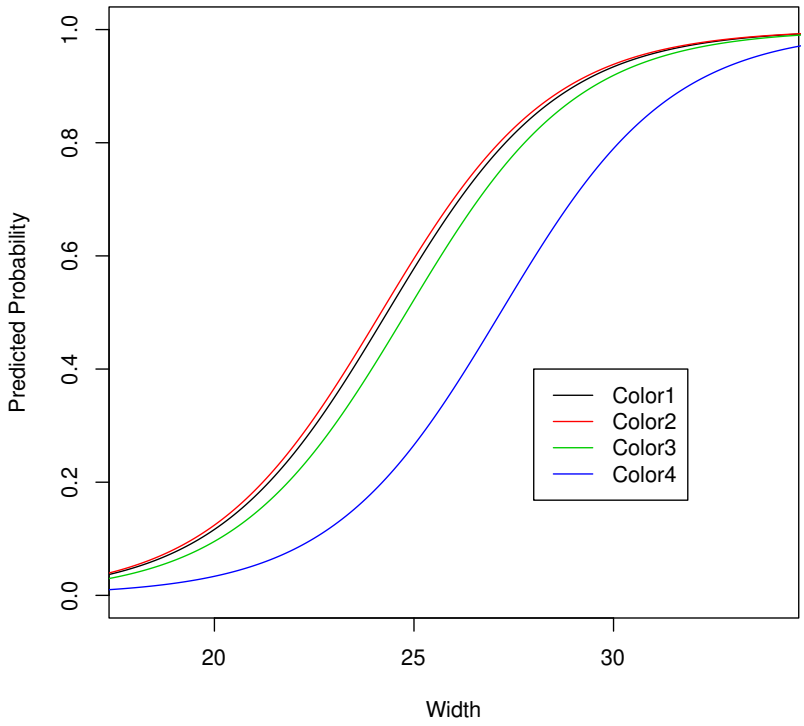
```
library(cdabookdb)
library(dplyr)
data("horseshoecrabs")
horseshoecrabs <- horseshoecrabs %>%
  mutate(
    Color_factor = factor(Color, 4:1),  # 将 Color 转换为因子，并设置因子水平
    psat = as.integer(horseshoecrabs$Satellites > 0)  # psat 为是否有追随者
  )


m1 <- glm(
  psat ~ Width + Color_factor, data = horseshoecrabs, family = binomial()
)
summary(m1)
```

```
##
## Call:
## glm(formula = psat ~ Width + Color_factor, family = binomial(),
##     data = horseshoecrabs)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.112  -0.985   0.524   0.851   2.141
##
## Coefficients:
```

```
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -12.715      2.762   -4.60  4.1e-06 ***
## Width             0.468      0.106    4.43  9.3e-06 ***
## Color_factor3     1.106      0.592    1.87    0.062 .
## Color_factor2     1.402      0.548    2.56    0.011 *
## Color_factor1     1.330      0.853    1.56    0.119
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 225.76  on 172  degrees of freedom
## Residual deviance: 187.46  on 168  degrees of freedom
## AIC: 197.5
##
## Number of Fisher Scoring iterations: 4
```

模型的详细解释可从教材中得到。以下画出了四种颜色下预测概率与宽度的关系曲线（教材图 4.4）

```
# 画出空图
plot(
  NULL,  # 不画任何点或线，只画出一个空图，供之后添加曲线使用
  xlim = c(18, 34), ylim = c(0, 1),  # 横纵坐标范围
  xlab = "Width", ylab = "Predicted Probability"  # 横纵坐标标签
)

sapply(1:4, function(i) {
  newdata <- data.frame(
    Width = seq(17, 35, 0.1),
    Color_factor = as.character(i)
  )
  pred_prop <- predict(m1, newdata, type = "response")  # 计算预测概率
  points(newdata$Width, pred_prop, type = "l", col = i)  # 绘制曲线
})

legend(28, 0.4, col = 1:4, legend = paste0("Color", 1:4), lty = 1)  # 图例
```



接着考虑 4.4.3 节中的有序预测变量的处理。此节中的案例与 4.4.1 节类似，但此处颜色变量不再是因子型，而是颜色得分。此处得分与数据集中一致，因此不必做额外处理，可直接回归。

```
m2 <- glm(psat ~ Color + Width, family = binomial(), data = horseshoecrabs)
summary(m2)
```

```
##
## Call:
## glm(formula = psat ~ Color + Width, family = binomial(), data = horseshoecrabs)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.169  -0.989   0.543   0.870   1.974
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -10.071      2.807   -3.59  0.00033 ***
## Color         -0.509      0.224   -2.28  0.02286 *
## Width          0.458      0.104    4.41  1.1e-05 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 225.76  on 172  degrees of freedom
## Residual deviance: 189.12  on 170  degrees of freedom
## AIC: 195.1
##
## Number of Fisher Scoring iterations: 4
```

而 4.4.4 节引入了交互效应。在拟合该模型前需要按教材中说明构造出一个颜色是否为深色的哑变量，之后再拟合包含交互效应的模型。

```
horseshoecrabs$is_dark <- as.character(horseshoecrabs$Color < 4)
# is_dark * Width 表示包含交互项以及 is_dark 和 Width 两个变量
# 若只想包含交互项应使用 is_dark:Width
m3 <- glm(
  psat ~ is_dark * Width,
  family = binomial(),
  data = horseshoecrabs
)
summary(m3)
```

```
##
## Call:
## glm(formula = psat ~ is_dark * Width, family = binomial(), data = horseshoecrabs)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.136  -0.934   0.500   0.855   1.775
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -5.854      6.694   -0.87     0.38
## is_darkTRUE         -6.958      7.318   -0.95     0.34
## Width                0.200      0.262    0.77     0.44
## is_darkTRUE:Width    0.322      0.286    1.13     0.26
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 225.76  on 172  degrees of freedom
## Residual deviance: 186.79  on 169  degrees of freedom
## AIC: 194.8
##
## Number of Fisher Scoring iterations: 4
```

4.5 logistic 回归效应的概括

课后题

第 8 题

(a)

```
library(cdabookdb)
data("horseshoecrabs")
horseshoecrabs$psat <- as.integer(horseshoecrabs$Satellites > 0)
m_crab <- glm(psat ~ Weight, data = horseshoecrabs, family = binomial())
summary(m_crab)
```

```
##
## Call:
## glm(formula = psat ~ Weight, family = binomial(), data = horseshoecrabs)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.111  -1.075   0.543   0.912   1.629
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -3.695      0.880   -4.20  2.7e-05 ***
## Weight         1.815      0.377    4.82  1.4e-06 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 225.76  on 172  degrees of freedom
## Residual deviance: 195.74  on 171  degrees of freedom
## AIC: 199.7
##
## Number of Fisher Scoring iterations: 4
```

从而模型为 $\text{logit}(\pi) = 3.6947 + 1.8151 Weight$

(b)

```
predict(m_crab, data.frame(Weight = c(1.2, 2.44, 5.2)), type = "response")
```

```
##      1      2      3
## 0.1800 0.6757 0.9968
```

从而这三种重量的母蛏有追随者的概率分别为 18.00%, 67.57%, 99.68%

(c)

$$\pi = \frac{e^{\alpha+\beta x}}{1+e^{\alpha+\beta x}} = 0.5$$
$$e^{\alpha+\beta x} = 1$$
$$\alpha + \beta x = 0$$
$$x = -\frac{\alpha}{\beta} = 2.0355$$

(d)

i). $\hat{\beta}\pi(1-\pi) = 0.25 \times 1.8145 = 0.4536$

ii). $0.1 \times 0.4536 = 0.0454$

ii). $0.58 \times 0.4536 = 0.2631$

(e) $\hat{\beta}$ 的 95% 置信区间为 $[1.8151 - 1.96 \times 0.3767, 1.8151 + 1.96 \times 0.3767] = [1.0768, 2.5534]$

优势比的 95% 置信区间为 $[e^{1.0768}, e^{2.5534}] = [2.9352, 12.8511]$

可以发现，有追随者的母蛏的重量明显比没有追随者的母蛏的重量大得多

(f)

$$z^2 = \left(\frac{1.8151}{0.3767}\right)^2 = 21.2172$$

而 $df = 1$，$pvalue < 0.0001$

从而确实存在重量的影响。

第 24 题

(a)

```
library(cdabookdb)
data("throat")
m_throat <- glm(Y ~ D + factor(T), data = throat, family = binomial())
summary(m_throat)
```

```
##
## Call:
## glm(formula = Y ~ D + factor(T), family = binomial(), data = throat)
```

```
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.380  -0.536   0.305   0.731   1.782
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.4173     1.0946   -1.29   0.1954
## D             0.0687     0.0264    2.60   0.0093 **
## factor(T)1   -1.6589     0.9229   -1.80   0.0722 .
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 46.180  on 34  degrees of freedom
## Residual deviance: 30.138  on 32  degrees of freedom
## AIC: 36.14
##
## Number of Fisher Scoring iterations: 5
```

从而模型为

$$\text{logit}(\pi) = 1.417 + 0.069D1.659T$$

模型表明，当控制其他变量不变时：

- 当 D 增加 1 时，$Y = 1$ 的优势会变为原来的 $e^{0.069} = 1.0711$ 倍
- $T = 1$ 和 $T = 0$ 的优势比为 $e^{-1.659} = 0.1903$

(b) 根据 R 输出结果，$\hat{\beta}_D$ 的 p 值为 $0.009 < 0.01$。所以可以认为存在 D 的影响。

(c)

```
m0_throat <- glm(Y ~ D * factor(T), data = throat, family = binomial())
summary(m0_throat)
```

```
##
## Call:
## glm(formula = Y ~ D * factor(T), family = binomial(), data = throat)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.971  -0.378   0.345   0.729   1.996
##
```

```
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)     0.0498     1.4694    0.03     0.97
## D               0.0285     0.0343    0.83     0.41
## factor(T)1     -4.4722     2.4671   -1.81     0.07 .
## D:factor(T)1    0.0746     0.0578    1.29     0.20
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 46.180  on 34  degrees of freedom
## Residual deviance: 28.321  on 31  degrees of freedom
## AIC: 36.32
##
## Number of Fisher Scoring iterations: 6
```

则当 $T=1$ 时，

$$\text{logit}(\pi) = 0.04984.4722 + 0.0285D + 0.0746D = 4.4224 + 0.1031D$$

当 $T=0$ 时，

$$\text{logit}(\pi) = 0.04749 + 0.0285D$$

对于 $T=1$ 的模型，当 D 增加 1 时，优势变为原来的 $e^{0.1031} = 1.1086$ 倍对于 $T=0$ 的模型，当 D 增加 1 时，优势变为原来的 $e^{0.0285} = 1.0289$ 倍

(d)

```
anova(m_throat, m0_throat, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: Y ~ D + factor(T)
## Model 2: Y ~ D * factor(T)
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        32       30.1
## 2        31       28.3  1     1.82     0.18
```

p 值为 $0.1777 > 0.05$，所以可以认为不需要交互项

第五章　logistic 回归模型的构建和应用

5.1　模型选择策略

母鲎及其追随者（模型选择）

该案例中，最开始的模型包含了重量、宽度、棘刺和颜色四个因素，其中棘刺和颜色是因子型变量。在 horseshoecrabs 数据集中这两者均为数值型，需要先进行转换。

```
library(cdabookdb)
library(dplyr)
data(horseshoecrabs)
horseshoecrabs <- horseshoecrabs %>%
  mutate(
    psat = as.integer(horseshoecrabs$Satellites > 0),  # psat 为是否有追随者
    Spine_factor = factor(Spine, levels = 3:1),  # 棘刺分组，棘刺 3 为基准
    Color_factor = factor(Color, levels = 4:1)  # 颜色分组，颜色 4 为基准
  )

m1 <- glm(
  psat ~ Weight + Width + Spine_factor + Color_factor,
  family = binomial(), data = horseshoecrabs
)
summary(m1)
```

```
##
## Call:
## glm(formula = psat ~ Weight + Width + Spine_factor + Color_factor,
##     family = binomial(), data = horseshoecrabs)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.198  -0.942   0.485   0.849   2.120
##
```

```
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -9.273      3.838   -2.42   0.0157 *
## Weight           0.826      0.704    1.17   0.2407
## Width            0.263      0.195    1.35   0.1779
## Spine_factor2   -0.496      0.629   -0.79   0.4302
## Spine_factor1   -0.400      0.503   -0.80   0.4259
## Color_factor3    1.120      0.593    1.89   0.0591 .
## Color_factor2    1.506      0.567    2.66   0.0079 **
## Color_factor1    1.609      0.936    1.72   0.0855 .
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 225.76  on 172  degrees of freedom
## Residual deviance: 185.20  on 165  degrees of freedom
## AIC: 201.2
##
## Number of Fisher Scoring iterations: 4
```

```
m2 <- glm(
  psat ~ Weight + Width + Spine_factor + Color_factor +
    Color_factor * Spine_factor + Width * Color_factor +
    Width * Spine_factor,
  family = binomial(),
  data = horseshoecrabs
)
summary(m2)
```

```
##
## Call:
## glm(formula = psat ~ Weight + Width + Spine_factor + Color_factor +
##     Color_factor * Spine_factor + Width * Color_factor + Width *
##     Spine_factor, family = binomial(), data = horseshoecrabs)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.176  -0.885   0.458   0.773   1.923
##
## Coefficients:
##                              Estimate Std. Error z value
```

```
## (Intercept)                 -6.48e-01   7.66e+00   -0.08
## Weight                       1.04e+00   7.54e-01    1.38
## Width                       -8.79e-02   3.26e-01   -0.27
## Spine_factor2               -1.72e+01   3.96e+03    0.00
## Spine_factor1               -1.80e+01   3.96e+03    0.00
## Color_factor3               -1.61e+01   1.00e+01   -1.61
## Color_factor2               -3.14e+00   8.85e+00   -0.35
## Color_factor1               -2.11e+01   3.96e+03   -0.01
## Spine_factor2:Color_factor3  1.63e+01   3.96e+03    0.00
## Spine_factor1:Color_factor3  3.34e+01   4.48e+03    0.01
## Spine_factor2:Color_factor2  1.57e+01   3.96e+03    0.00
## Spine_factor1:Color_factor2  1.69e+01   3.96e+03    0.00
## Spine_factor2:Color_factor1  5.27e+01   6.25e+03    0.01
## Spine_factor1:Color_factor1  3.61e+01   5.59e+03    0.01
## Width:Color_factor3          6.70e-01   3.94e-01    1.70
## Width:Color_factor2          1.84e-01   3.43e-01    0.54
## Width:Color_factor1          1.45e-01   7.88e-01    0.18
## Width:Spine_factor2          9.81e-03   6.70e-01    0.01
## Width:Spine_factor1          1.77e-02   2.89e-01    0.06
##                             Pr(>|z|)
## (Intercept)                    0.933
## Weight                         0.167
## Width                          0.788
## Spine_factor2                  0.997
## Spine_factor1                  0.996
## Color_factor3                  0.108
## Color_factor2                  0.723
## Color_factor1                  0.996
## Spine_factor2:Color_factor3    0.997
## Spine_factor1:Color_factor3    0.994
## Spine_factor2:Color_factor2    0.997
## Spine_factor1:Color_factor2    0.997
## Spine_factor2:Color_factor1    0.993
## Spine_factor1:Color_factor1    0.995
## Width:Color_factor3            0.089 .
## Width:Color_factor2            0.591
## Width:Color_factor1            0.854
## Width:Spine_factor2            0.988
## Width:Spine_factor1            0.951
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 225.76  on 172  degrees of freedom
## Residual deviance: 171.66  on 154  degrees of freedom
## AIC: 209.7
##
## Number of Fisher Scoring iterations: 16
```

```
anova(m2)
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: psat
##
## Terms added sequentially (first to last)
##
##
##                           Df Deviance Resid. Df Resid. Dev
## NULL                                        172        226
## Weight                     1    30.02       171        196
## Width                      1     2.85       170        193
## Spine_factor               2     0.09       168        193
## Color_factor               3     7.60       165        185
## Spine_factor:Color_factor  6     9.61       159        176
## Width:Color_factor         3     3.93       156        172
## Width:Spine_factor         2     0.00       154        172
```

```
# 向后逐步回归
m2_backward <- step(m2, direction = "backward", trace = FALSE)
m2_backward
```

```
##
## Call:  glm(formula = psat ~ Width + Color_factor, family = binomial(),
##     data = horseshoecrabs)
##
## Coefficients:
##   (Intercept)          Width  Color_factor3  Color_factor2
##       -12.715          0.468          1.106          1.402
## Color_factor1
##         1.330
##
```

```
## Degrees of Freedom: 172 Total (i.e. Null);  168 Residual
## Null Deviance:       226
## Residual Deviance: 187   AIC: 197
```

```
# 双向逐步回归
m2_step <- step(m2, trace = FALSE)
summary(m2_step)
```

```
##
## Call:
## glm(formula = psat ~ Width + Color_factor, family = binomial(),
##     data = horseshoecrabs)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.112  -0.985   0.524   0.851   2.141
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -12.715      2.762   -4.60  4.1e-06 ***
## Width            0.468      0.106    4.43  9.3e-06 ***
## Color_factor3    1.106      0.592    1.87    0.062 .
## Color_factor2    1.402      0.548    2.56    0.011 *
## Color_factor1    1.330      0.853    1.56    0.119
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 225.76  on 172  degrees of freedom
## Residual deviance: 187.46  on 168  degrees of freedom
## AIC: 197.5
##
## Number of Fisher Scoring iterations: 4
```

母蚕及其追随者（预测功效）

首先得到模型

```
library(cdabookdb)
data("horseshoecrabs")
```

```
horseshoecrabs$psat <- as.integer(horseshoecrabs$Satellites > 0)
m <- glm(
  psat ~ factor(Color) + Width,
  data = horseshoecrabs, family = binomial()
)
```

然后就可以获取混淆矩阵（交叉分类表）

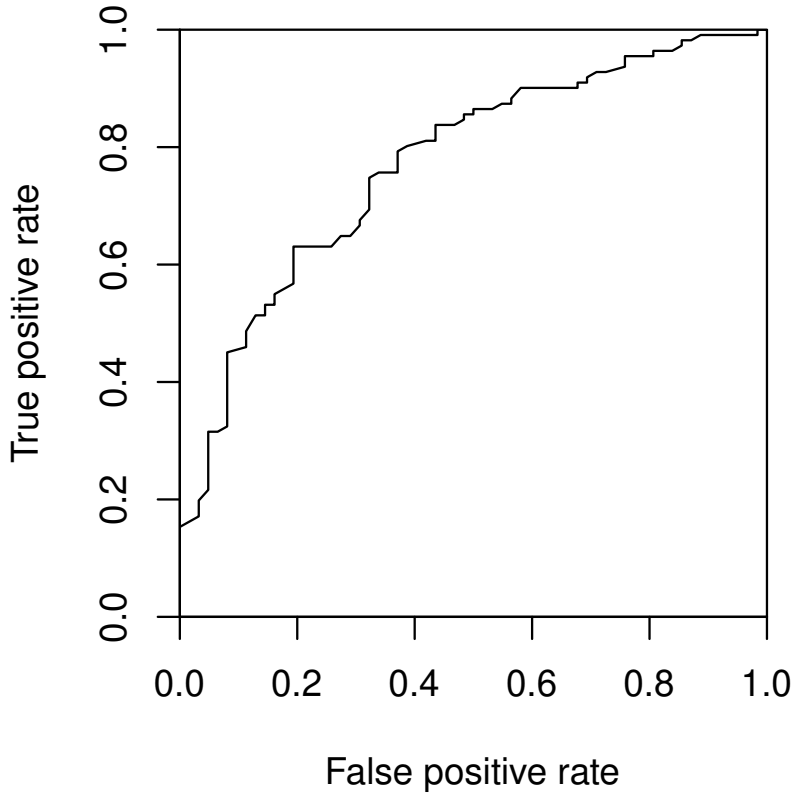
```
pi0 <- 0.5  # cut-off value
pred_prob <- predict(m, type = "response")
pred_type <- cut(
  pred_prob, breaks = c(0, pi0, 1), labels = 0:1,
  include.lowest = TRUE
)
table(horseshoecrabs$psat, pred_type)
```

```
##    pred_type
##       0  1
##   0 31 31
##   1 15 96
```

这里结果和书上不大一样，原因不明

画 ROC 曲线和计算 AUC 可以使用 `ROCR` 包中的 `performance`

```
library(ROCR)
par(pty = "s")
pred <- prediction(fitted(m2_step), horseshoecrabs$psat)
perf <- performance(pred, "tpr", "fpr")
plot(perf, asp =1, xaxs="i", yaxs="i")
```

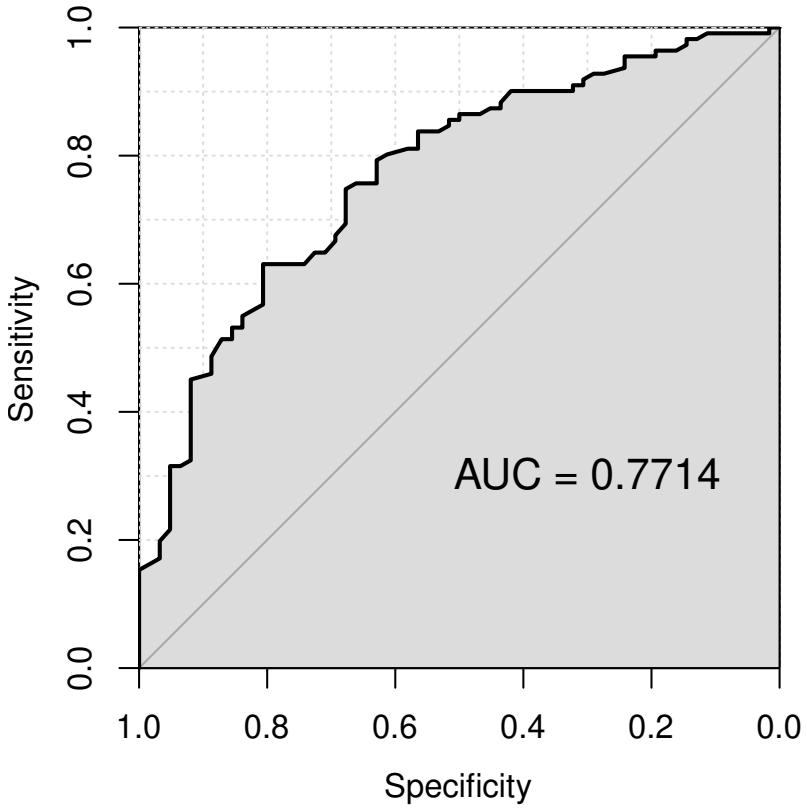



```
performance(pred,"auc")@y.values[[1]]
```

```
## [1] 0.7714
```

或者也可以通过 pROC 包的 roc 函数，有更多可调节的作图选项（可查看 help(plot.roc)）

```
library(pROC)
par(pty = "s")
result <- roc(
  horseshoecrabs$psat,
  predict(m2_step, type = "response"),
  plot = TRUE,
  auc.polygon = TRUE,
  grid = TRUE,
  asp =1,
  xaxs="i",
  yaxs="i"
)
text(0.3, 0.3, labels = paste("AUC =", round(result$auc, 4)), cex = 1.3)
```



最后是计算真实分类与预测概率的相关性

```
cor(horseshoecrabs$psat, fitted(m))
```

```
## [1] 0.4522
```

5.2 模型检验

母蛏及其追随者（模型 LR 检验）

以下是对是否有需要宽度的二次项进行的 LR 检验

```
library(cdabookdb)
data("horseshoecrabs")
# 分别拟合出没有二次项和有二次项的模型
m1 <- glm(
  Satellites > 0 ~ Width,
  data = horseshoecrabs, family = binomial()
)
m2 <- glm(
  Satellites > 0 ~ Width + I(Width ^ 2),
  data = horseshoecrabs, family = binomial()
```

```
)

# 查看二次项系数
summary(m2)
```

```
##
## Call:
## glm(formula = Satellites > 0 ~ Width + I(Width^2), family = binomial(),
##     data = horseshoecrabs)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.119  -1.044   0.507   0.948   1.541
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  14.5916    30.2237    0.48     0.63
## Width        -1.5957     2.3520   -0.68     0.50
## I(Width^2)    0.0405     0.0457    0.89     0.38
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 225.76  on 172  degrees of freedom
## Residual deviance: 193.63  on 170  degrees of freedom
## AIC: 199.6
##
## Number of Fisher Scoring iterations: 5
```

```
# 对比两个模型（似然比检验）
anova(m1, m2, test = "LR")
```

```
## Analysis of Deviance Table
##
## Model 1: Satellites > 0 ~ Width
## Model 2: Satellites > 0 ~ Width + I(Width^2)
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       171        194
## 2       170        194  1    0.825     0.36
```

AZT 和 AIDS（拟合优度）

对列联表进行 logistic 回归有两种方法，一种是转换为数据框进行回归，使用 `weights=data$Freq` 设定次数，另一种是直接使用列联表的表格进行回归。而要进行 X2 和 G2 的拟合优度检验，最好使用第二种方法

```
library(cdabookdb)
library(tidyr)
data("AZT")
AZT_df <- spread(as.data.frame(AZT), Symptoms, Freq)
AZT_df
```

```
##    Race AZTUse Yes No
## 1 White    Yes  14 93
## 2 White     No  32 81
## 3 Black    Yes  11 52
## 4 Black     No  12 43
```

```
m <- glm(
  cbind(Yes, No) ~ (Race == "White") + (AZTUse == "Yes"),
  data = AZT_df,
  family = binomial("logit")
)
summary(m)
```

```
##
## Call:
## glm(formula = cbind(Yes, No) ~ (Race == "White") + (AZTUse ==
##     "Yes"), family = binomial("logit"), data = AZT_df)
##
## Deviance Residuals:
##      1       2       3       4
## -0.555   0.425   0.704  -0.633
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -1.0736     0.2629   -4.08  4.4e-05
## Race == "White"TRUE   0.0555     0.2886    0.19   0.8476
## AZTUse == "Yes"TRUE  -0.7195     0.2790   -2.58   0.0099
##
## (Intercept)         ***
## Race == "White"TRUE
## AZTUse == "Yes"TRUE **
```

```
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 8.3499  on 3  degrees of freedom
## Residual deviance: 1.3835  on 1  degrees of freedom
## AIC: 24.86
##
## Number of Fisher Scoring iterations: 4
```

```
# X2 和 G2 的自由度
df <- nrow(AZT_df) - length(coef(m))

# X2 检验
X2 <- sum(resid(m, type = "pearson") ^ 2)
x2_pvalue <- 1- pchisq(X2, df)
c(X2 = X2, pvalue = x2_pvalue)
```

```
##     X2 pvalue
## 1.3910 0.2382
```

```
# G2 检验
G2 <- sum(resid(m, type = "deviance") ^ 2)
g2_pvalue <- 1 - pchisq(G2, df)
c(G2 = G2, pvalue = g2_pvalue)
```

```
##     G2 pvalue
## 1.3835 0.2395
```

母蛏及其追随者（HM 检验）

Hosmer–Lemeshow 检验可使用 `ResourceSelection` 包中的 `hoslem.test()` 来得到

```
library(cdabookdb)
library(ResourceSelection)
data("horseshoecrabs")
horseshoecrabs$psat <- as.integer(horseshoecrabs$Satellites > 0)
m <- glm(
  psat ~ factor(Color) + Width,
  data = horseshoecrabs, family = binomial()
```

```
)

# Hosmer-Lemeshow test
hoslem.test(m$y, fitted(m))
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  m$y, fitted(m)
## X-squared = 4.5, df = 8, p-value = 0.8
```

佛罗里达大学研究生入学

```
library(cdabookdb)
library(tidyr)
data("UFAdmissions")
UFAdmissions_df <- spread(as.data.frame(UFAdmissions), Decision, Freq)
UFAdmissions_df
```

```
##    Dept Gender Admitted Rejected
## 1  anth Female       32       81
## 2  anth   Male       21       41
## 3  astr Female        6        0
## 4  astr   Male        3        8
## 5  chem Female       12       43
## 6  chem   Male       34      110
## 7  clas Female        3        1
## 8  clas   Male        4        0
## 9  comm Female       52      149
## 10 comm   Male        5       10
## 11 comp Female        8        7
## 12 comp   Male        6       12
## 13 engl Female       35      100
## 14 engl   Male       30      112
## 15 geog Female        9        1
## 16 geog   Male       11       11
## 17 geol Female        6        3
## 18 geol   Male       15        6
## 19 germ Female       17        0
## 20 germ   Male        4        1
## 21 hist Female        9        9
```

```
## 22 hist   Male       21       19
## 23 lati Female       26        7
## 24 lati   Male       25       16
## 25 ling Female       21       10
## 26 ling   Male        7        8
## 27 math Female       25       18
## 28 math   Male       31       37
## 29 phil Female        3        0
## 30 phil   Male        9        6
## 31 phys Female       10       11
## 32 phys   Male       25       53
## 33 poli Female       25       34
## 34 poli   Male       39       49
## 35 psyc Female        2      123
## 36 psyc   Male        4       41
## 37 reli Female        3        3
## 38 reli   Male        0        2
## 39 roma Female       29       13
## 40 roma   Male        6        3
## 41 soci Female       16       33
## 42 soci   Male        7       17
## 43 stat Female       23        9
## 44 stat   Male       36       14
## 45 zool Female        4       62
## 46 zool   Male       10       54
```

```
m <- glm(
  cbind(Admitted, Rejected) ~ Dept,
  data = UFAdmissions_df,
  family = binomial()
)
summary(m)
```

```
##
## Call:
## glm(formula = cbind(Admitted, Rejected) ~ Dept, family = binomial(),
##     data = UFAdmissions_df)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.728  -0.653  -0.001   0.762   2.763
##
## Coefficients:
```

```
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.8337     0.1645   -5.07  4.0e-07 ***
## Deptastr      0.9515     0.5130    1.85  0.06363 .
## Deptchem     -0.3681     0.2352   -1.56  0.11767
## Deptclas      2.7796     1.0816    2.57  0.01017 *
## Deptcomm     -0.1921     0.2256   -0.85  0.39444
## Deptcomp      0.5283     0.3887    1.36  0.17411
## Deptengl     -0.3485     0.2172   -1.60  0.10860
## Deptgeog      1.3446     0.4005    3.36  0.00079 ***
## Deptgeol      1.6810     0.4310    3.90  9.6e-05 ***
## Deptgerm      3.8783     1.0367    3.74  0.00018 ***
## Depthist      0.9027     0.3100    2.91  0.00359 **
## Deptlati      1.6301     0.3003    5.43  5.7e-08 ***
## Deptling      1.2756     0.3440    3.71  0.00021 ***
## Deptmath      0.8517     0.2512    3.39  0.00070 ***
## Deptphil      1.5269     0.5264    2.90  0.00372 **
## Deptphys      0.2302     0.2669    0.86  0.38851
## Deptpoli      0.5738     0.2340    2.45  0.01419 *
## Deptpsyc     -2.4744     0.4470   -5.54  3.1e-08 ***
## Deptreli      0.3229     0.7486    0.43  0.66622
## Deptroma      1.6165     0.3437    4.70  2.6e-06 ***
## Deptsoci      0.0572     0.3009    0.19  0.84923
## Deptstat      1.7758     0.2958    6.00  1.9e-09 ***
## Deptzool     -1.2808     0.3273   -3.91  9.1e-05 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 449.830  on 45  degrees of freedom
## Residual deviance:  44.735  on 23  degrees of freedom
## AIC: 241.4
##
## Number of Fisher Scoring iterations: 5
```

```r
# X2 和 G2 的自由度
df <- nrow(UFAdmissions_df) - length(coef(m))

# X2 检验
X2 <- sum(resid(m, type = "pearson") ^ 2)
x2_pvalue <- 1- pchisq(X2, df)
```

```
c(X2 = X2, pvalue = x2_pvalue)
```

```
##       X2   pvalue
## 40.85236  0.01231
```

```
# G2 检验
G2 <- sum(resid(m, type = "deviance") ^ 2)
g2_pvalue <- 1 - pchisq(G2, df)
c(G2 = G2, pvalue = g2_pvalue)
```

```
##        G2    pvalue
## 44.735165  0.004282
```

心脏病与血压的关系

```
library(cdabookfunc)
library(cdabookdb)
data("blood_pressure")
m <- glm(
  cbind(ObservedDisease, SampleSize - ObservedDisease) ~ BloodPressure,
  data = blood_pressure,
  family = binomial()
)
summary(m)
```

```
##
## Call:
## glm(formula = cbind(ObservedDisease, SampleSize - ObservedDisease) ~
##     BloodPressure, family = binomial(), data = blood_pressure)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.062  -0.598  -0.225   0.214   1.850
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -6.08203    0.72432   -8.40   <2e-16 ***
## BloodPressure  0.02434    0.00484    5.03    5e-07 ***
## ---
## Signif. codes:
```

```
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 30.0226  on 7  degrees of freedom
## Residual deviance:  5.9092  on 6  degrees of freedom
## AIC: 42.61
##
## Number of Fisher Scoring iterations: 4
```

关于书上表 5.6 的计算，其中 `Dfbeta` 这一项与 R 函数 `dfbeta()` 得到的结果有些许差异，是因为这个表是使用 SAS 计算得到的，而 SAS 计算 `Dfbeta` 的方式与 R 的不同。SAS 的计算方法详见 SAS 关于 logistic 回归诊断的说明文档[1]

此外表中还有一些变量在 R 中没法直接计算，比如 `c` 和 `LR Difference` 等，这些变量在以上说明文档中也有相应的定义。

而要使用 SAS 的方法计算以上这些变量,我在 `cdabookcode` 中定义了 `dfbetas_logit_sas()` 和 `influence_logit_sas()` 这两个函数，前一个使用了 SAS 的方法计算 `Dfbetas`，而后一个计算了以上 SAS 说明文档中列出的所有诊断统计量。

```
# 对比 R 和 SAS 的 DFBETAS
dfbetas_compare <- data.frame(
  R = dfbetas(m),
  SAS = dfbetas_logit_sas(m)
)
xtable::xtable(dfbetas_compare, align = "ccccc", digits = 2)
```

R..Intercept.	R.BloodPressure	SAS..Intercept.	SAS.BloodPressure
-0.61	0.56	-0.53	0.49
2.50	-2.24	1.28	-1.14
-0.41	0.34	-0.39	0.33
-0.12	0.08	-0.12	0.08
-0.00	0.01	-0.00	0.01
0.05	-0.06	0.05	-0.07
-0.33	0.38	-0.35	0.40
0.10	-0.11	0.11	-0.12

```
# 计算所有诊断统计量
result <- influence_logit_sas(m, "data.frame")
result$`dfbetas..Intercept.` <- NULL
names(result) <- c(
```

[1] https://support.sas.com/documentation/cdl/en/statug/63347/HTML/default/viewer.htm#statug_logistic_sect049.htm

```
  "hat", "pearson", "deviance", "dfbetas",
  "c", "cbar", "difchisq", "difdev"
)
xtable::xtable(result, align = "ccccccccc", digits = 2)
```

hat	pearson	deviance	dfbetas	c	cbar	difchisq	difdev
0.22	-0.98	-1.06	0.49	0.34	0.26	1.22	1.39
0.29	2.01	1.85	-1.14	2.26	1.62	5.64	5.04
0.26	-0.81	-0.84	0.33	0.31	0.23	0.89	0.94
0.22	-0.51	-0.52	0.08	0.09	0.07	0.33	0.34
0.13	0.12	0.12	0.01	0.00	0.00	0.02	0.02
0.13	-0.30	-0.31	-0.07	0.02	0.01	0.11	0.11
0.38	0.51	0.50	0.40	0.26	0.16	0.43	0.42
0.38	-0.14	-0.14	-0.12	0.02	0.01	0.03	0.03

```
# 标准化 pearson 残差
round(rstandard(m, type = "pearson"), 2)
```

```
##     1     2     3     4     5     6     7     8
## -1.11  2.37 -0.95 -0.57  0.13 -0.33  0.65 -0.18
```

5.3 稀疏数据效应

稀疏数据的临床试验结果

```
library(cdabookdb)
library(dplyr)
data("treatment3")
treatment3_df1 <- as.data.frame(treatment3)
treatment3_df1$Center <- factor(treatment3_df1$Center, 5:1)
treatment3_df2 <- spread(treatment3_df1, Response, Freq)


# 使用数据框进行回归
m1_df1 <- glm(
  (Response == "Success") ~ Center + Treatment,
  family = binomial(), weights = Freq,
  data = treatment3_df1
)
```

```
summary(m1_df1)
```

```
##
## Call:
## glm(formula = (Response == "Success") ~ Center + Treatment, family = binomial(),
##     data = treatment3_df1, weights = Freq)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9488  -0.7277  -0.0001   0.5665   3.0974
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -0.476      0.506   -0.94    0.346
## Center4             1.063      0.701    1.52    0.129
## Center3           -18.614   2985.252   -0.01    0.995
## Center2            -2.180      1.133   -1.92    0.054 .
## Center1           -18.587   3180.370   -0.01    0.995
## TreatmentPlacebo   -1.546      0.702   -2.20    0.028 *
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 85.77  on 14  degrees of freedom
## Residual deviance: 57.74  on  9  degrees of freedom
## AIC: 69.74
##
## Number of Fisher Scoring iterations: 17
```

```
# 使用列联表进行回归
m1_df2 <- glm(
  cbind(Success, Failure) ~ Center + Treatment,
  family = binomial(),
  data = treatment3_df2
)
summary(m1_df2)
```

```
##
## Call:
## glm(formula = cbind(Success, Failure) ~ Center + Treatment, family = binomial(),
```

```
##     data = treatment3_df2)
##
## Deviance Residuals:
##      1       2       3       4       5       6       7
## -0.201   0.294   0.151  -0.173   0.000   0.000   0.161
##      8       9      10
## -0.545   0.000   0.000
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -0.476      0.506   -0.94    0.346
## Center4              1.063      0.701    1.52    0.129
## Center3            -22.565  21523.645    0.00    0.999
## Center2             -2.180      1.133   -1.92    0.054 .
## Center1            -22.570  23296.396    0.00    0.999
## TreatmentPlacebo    -1.546      0.702   -2.20    0.028 *
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 28.53202  on 9  degrees of freedom
## Residual deviance:  0.50214  on 4  degrees of freedom
## AIC: 24.86
##
## Number of Fisher Scoring iterations: 21
```

两个模型治疗中心 1 和治疗中心 3 的系数绝对值和 SE 都很大，并且在两个模型中的系数是不同的。而其他变量则正常，并且在两个模型有相同的系数和 SE。

而接下来去除截距项，重新拟合。

```
m2_df1 <- glm(
  (Response == "Success") ~ Center + Treatment - 1,
  family = binomial(), weights = Freq,
  data = treatment3_df1
)
summary(m2_df1)
```

```
##
## Call:
## glm(formula = (Response == "Success") ~ Center + Treatment -
##     1, family = binomial(), data = treatment3_df1, weights = Freq)
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9488  -0.7277  -0.0001   0.5665   3.0974
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## Center5            -0.476      0.506   -0.94    0.346
## Center4             0.587      0.605    0.97    0.332
## Center3           -19.090   2985.252   -0.01    0.995
## Center2            -2.657      1.036   -2.56    0.010 *
## Center1           -19.064   3180.370   -0.01    0.995
## TreatmentPlacebo   -1.546      0.702   -2.20    0.028 *
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 130.31  on 15  degrees of freedom
## Residual deviance:  57.74  on  9  degrees of freedom
## AIC: 69.74
##
## Number of Fisher Scoring iterations: 17
```

```r
m2_df2 <- glm(
  cbind(Success, Failure) ~ Center + Treatment - 1,
  family = binomial(),
  data = treatment3_df2
)
summary(m2_df2)
```

```
##
## Call:
## glm(formula = cbind(Success, Failure) ~ Center + Treatment -
##     1, family = binomial(), data = treatment3_df2)
##
## Deviance Residuals:
##      1       2       3       4       5       6       7
## -0.201   0.294   0.151  -0.173   0.000   0.000   0.161
##      8       9      10
## -0.545   0.000   0.000
##
```

```
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## Center5             -0.476      0.506   -0.94    0.346
## Center4              0.587      0.605    0.97    0.332
## Center3            -23.041  21523.645    0.00    0.999
## Center2             -2.657      1.036   -2.56    0.010 *
## Center1            -23.046  23296.396    0.00    0.999
## TreatmentPlacebo    -1.546      0.702   -2.20    0.028 *
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 73.07369  on 10  degrees of freedom
## Residual deviance:  0.50214  on  4  degrees of freedom
## AIC: 24.86
##
## Number of Fisher Scoring iterations: 21
```

结果与之前类似。

接着尝试不考虑治疗中心的效应

```
treatment3_margin <- margin.table(treatment3, c(2, 3))
treatment <- rownames(treatment3_margin)
treatment3_margin
```

```
##              Response
## Treatment     Success Failure
##   Active drug      12      36
##   Placebo           4      42
```

```
m3 <- glm(
  treatment3_margin ~ treatment,
  family = binomial()
)

summary(m3)
```

```
##
## Call:
## glm(formula = treatment3_margin ~ treatment, family = binomial())
##
```

```
## Deviance Residuals:
## [1]  0  0
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -1.099      0.333   -3.30  0.00098 ***
## treatmentPlacebo   -1.253      0.620   -2.02  0.04346 *
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance:  4.6054e+00  on 1  degrees of freedom
## Residual deviance: -5.3291e-15  on 0  degrees of freedom
## AIC: 11.23
##
## Number of Fisher Scoring iterations: 3
```

此时模型系数就正常了

5.4 条件 logistic 回归与精确推断

晋升能力

```
library(cdabookdb)
library(tidyr)
data("promotion_race")
promotion_race_df <- spread(as.data.frame(promotion_race), Promotion, Freq)

m <- glm(
  cbind(Yes, No) ~ Race + Month,
  data = promotion_race_df,
  family = binomial()
)

summary(m)
```

```
##
## Call:
## glm(formula = cbind(Yes, No) ~ Race + Month, family = binomial(),
```

```
##     data = promotion_race_df)
##
## Deviance Residuals:
##         1          2          3          4          5
## -9.52e-06  -1.06e-05  -7.98e-06  -4.20e-08   0.00e+00
##         6
##  0.00e+00
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -25.764  52607.802    0.00     1.00
## RaceWhite         24.377  52607.802    0.00     1.00
## MonthAugust        0.208      0.800    0.26     0.80
## MonthSeptember    -0.486      0.943   -0.51     0.61
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 8.2664e+00  on 5  degrees of freedom
## Residual deviance: 2.6585e-10  on 2  degrees of freedom
## AIC: 16.52
##
## Number of Fisher Scoring iterations: 23
```

模型中种族效应的估计值是一个非常极端的结果 (-24.38)

5.5 logistic 回归的样本量与功效

样本量计算

计算比较两个比例所需要样本量可以使用 `cdabookcode` 中的 `samplesize_prop()` 计算。

```
library(cdabookfunc)
library(cdabookdb)
samplesize_prop(0.2, 0.3, 0.05, 0.1)
```

```
## [1] 389
```

课后题

第 10 题

(a)

```
library(dplyr)
library(cdabookdb)
data("horseshoecrabs")
horseshoecrabs$psat <- horseshoecrabs$Satellites > 0
m1 <- glm(
  psat ~ Weight,
  data = horseshoecrabs,
  family = binomial()
)
summary(m1)
```

```
##
## Call:
## glm(formula = psat ~ Weight, family = binomial(), data = horseshoecrabs)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.111  -1.075   0.543   0.912   1.629
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -3.695      0.880   -4.20  2.7e-05 ***
## Weight         1.815      0.377    4.82  1.4e-06 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 225.76  on 172  degrees of freedom
## Residual deviance: 195.74  on 171  degrees of freedom
## AIC: 199.7
##
## Number of Fisher Scoring iterations: 4
```

```
# 预测类别
pred_type <- fitted(m1) > mean(horseshoecrabs$psat)
# 真实类别
true_type <- horseshoecrabs$psat
# 混淆矩阵
table(true_type, pred_type)
```

```
##          pred_type
## true_type FALSE TRUE
##     FALSE    45   17
##     TRUE     43   68
```

```
# 敏感度与特异度
table(true_type, pred_type) %>%
  prop.table(margin = 1) %>%
  round(4)
```

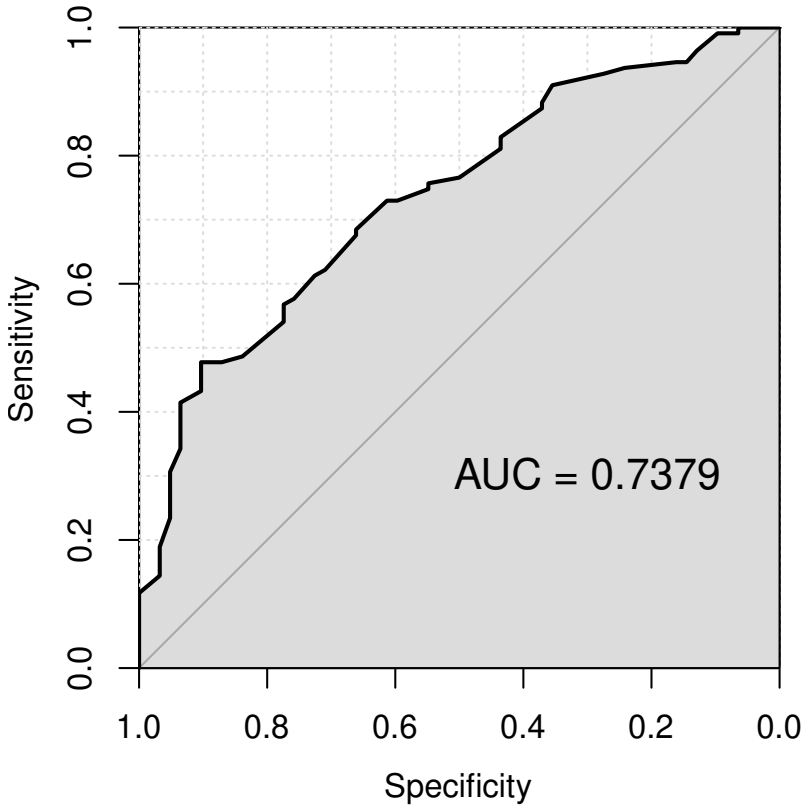
```
##          pred_type
## true_type  FALSE   TRUE
##     FALSE 0.7258 0.2742
##     TRUE  0.3874 0.6126
```

则模型的敏感度为 0.6126，特异度为 0.7258。

对于有追随者的母鲨，模型有 0.6126 的概率预测其有追随者；对于没有追随者的母鲨，模型有 0.7258 的概率预测其没有追随者；

(b)

```
library(pROC)
par(pty = "s")
result <- roc(
  true_type,
  fitted(m1),
  plot = TRUE,
  auc.polygon = TRUE,
  grid = TRUE,
  asp =1,
  xaxs="i",
  yaxs="i"
)
text(0.3, 0.3, labels = paste("AUC =", round(result$auc, 4)), cex = 1.3)
```



AUC 值为 0.7379

(c)

```
library(ResourceSelection)
hoslem.test(m1$y, fitted(m1), g = 10)
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  m1$y, fitted(m1)
## X-squared = 7.8, df = 8, p-value = 0.4
```

p 值为 0.4499，大于 0.05。因此我们认为模型是充分的。

(d)

```
m2 <- glm(
  psat ~ Weight + I(Weight ^ 2),
  data = horseshoecrabs,
  family = binomial()
)
summary(m2)
```

```
##
```

```
## Call:
## glm(formula = psat ~ Weight + I(Weight^2), family = binomial(),
##     data = horseshoecrabs)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.183  -1.074   0.520   0.939   1.543
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -1.888      3.549   -0.53     0.59
## Weight         0.218      3.082    0.07     0.94
## I(Weight^2)    0.339      0.654    0.52     0.60
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 225.76  on 172  degrees of freedom
## Residual deviance: 195.46  on 170  degrees of freedom
## AIC: 201.5
##
## Number of Fisher Scoring iterations: 5
```

(e)

```
c(m1 = AIC(m1), m2 = AIC(m2))
```

```
##    m1    m2
## 199.7 201.5
```

模型 1 有更小的 AIC 值，因此我们认为模型 1 更好，即不需要平方项。

第 18 题

(a)

```
library(cdabookdb)
library(tidyr)
data("smoking_lungcancer_cn")
ftable(smoking_lungcancer_cn)
```

```
##                      Smoking Yes  No
## City      LungCancer
## Beijing   Yes                126  35
```

```
##           No                 100  61
## Shanghai  Yes                908 497
##           No                 688 807
## Shenyang  Yes                913 336
##           No                 747 598
## Nanjing   Yes                235  58
##           No                 172 121
## Harbin    Yes                402 121
##           No                 308 215
## Zhengzhou Yes                182  72
##           No                 156  98
## Taiyuan   Yes                 60  11
##           No                  99  43
## Nanchang  Yes                104  21
##           No                  89  36
```

```
smoking_lungcancer_cn <- spread(
  as.data.frame(smoking_lungcancer_cn), LungCancer, Freq
)

m1 <- glm(
  cbind(Yes, No) ~ City + Smoking,
  data = smoking_lungcancer_cn,
  family = binomial()
)
summary(m1)
```

```
##
## Call:
## glm(formula = cbind(Yes, No) ~ City + Smoking, family = binomial(),
##     data = smoking_lungcancer_cn)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.2178  -0.1484  -0.0001   0.1682   1.3547
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)    0.22838    0.11398    2.00    0.045 *
## CityShanghai   0.05562    0.11957    0.47    0.642
## CityShenyang  -0.02774    0.12007   -0.23    0.817
## CityNanjing    0.00576    0.14091    0.04    0.967
## CityHarbin     0.01819    0.12947    0.14    0.888
```

```
## CityZhengzhou  0.02878    0.14476    0.20    0.842
## CityTaiyuan   -0.74568    0.18552   -4.02  5.8e-05 ***
## CityNanchang  -0.05491    0.17100   -0.32    0.748
## SmokingNo     -0.77706    0.04677  -16.61  < 2e-16 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 310.8951  on 15  degrees of freedom
## Residual deviance:   5.1958  on  7  degrees of freedom
## AIC: 121
##
## Number of Fisher Scoring iterations: 3
```

(b)

```
# X2 检验
df <- nrow(smoking_lungcancer_cn) - length(coef(m1))
X2 <- sum(resid(m1, type = "pearson") ^ 2)
x2_pvalue <- 1- pchisq(X2, df)
c(X2 = X2, pvalue = x2_pvalue)
```

```
##     X2 pvalue
## 5.1999 0.6356
```

X2 检验统计量为 5.2，p 值为 0.6356，大于 0.05。因此我们认为模型是充分的。

(c)

```
rstandard(m1)
```

```
##         1         2         3         4         5         6
##  0.038865 -0.038875 -0.247104  0.247059  0.001264 -0.001264
##         7         8         9        10        11        12
##  1.486229 -1.497384  0.500428 -0.501198 -1.708291  1.697803
##        13        14        15        16
##  0.229398 -0.231070 -0.268310  0.267550
```

```
range(rstandard(m1))
```

```
## [1] -1.708  1.698
```

标准化残差在-1.7 和 1.7 之间，这残差范围是正常合理的。

第 28 题

```
library(cdabookdb)
# (a)
samplesize_prop(0.2, 0.3, 0.1, 0.2)
```

```
## [1] 229
```

```
# (b)(i)
samplesize_prop(0.2, 0.3, 0.1, 0.1)
```

```
## [1] 317
```

```
# (b)(ii)
samplesize_prop(0.2, 0.3, 0.05, 0.2)
```

```
## [1] 291
```

```
# (b)(iii)
samplesize_prop(0.2, 0.3, 0.05, 0.1)
```

```
## [1] 389
```

第六章　多类别 logit 模型

6.1　名义响应变量的 logit 模型

钝吻鳄食物选择

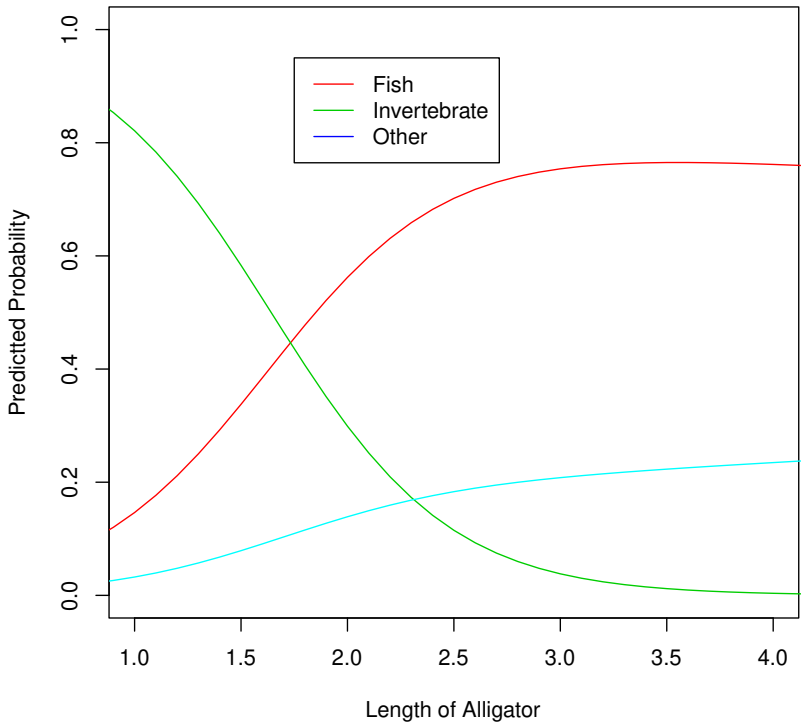
```
library(VGAM)
library(cdabookdb)
data("alligators1")

# 拟合多类别 logit 模型
alligators.fit1 <- vglm(
  Food ~ Length,
  family = multinomial,
  data=alligators1
)

summary(alligators.fit1)
```

```
##
## Call:
## vglm(formula = Food ~ Length, family = multinomial, data = alligators1)
##
##
## Pearson residuals:
##                       Min     1Q Median    3Q  Max
## log(mu[,1]/mu[,3]) -2.33 -0.507  0.554 0.684 1.45
## log(mu[,2]/mu[,3]) -2.69 -0.482 -0.165 0.709 3.44
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept):1    1.618      1.307    1.24   0.2159
## (Intercept):2    5.697      1.794    3.18   0.0015 **
```

```
## Length:1        -0.110      0.517   -0.21   0.8314
## Length:2        -2.465      0.900      NA       NA
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of linear predictors:  2
##
## Names of linear predictors:
## log(mu[,1]/mu[,3]), log(mu[,2]/mu[,3])
##
## Residual deviance: 98.34 on 114 degrees of freedom
##
## Log-likelihood: -49.17 on 114 degrees of freedom
##
## Number of iterations: 5
##
## Warning: Hauck-Donner effect detected in the following estimate(s):
## 'Length:2'
##
## Reference group is level  3  of the response
```

以下画出短吻鳄食用三种食物的概率随着其长度的变化曲线。

```
new_length_x <- data.frame(Length = seq(0, 5, 0.1))
prob_food <- predict(alligators.fit1, new_length_x, type = "response")

plot(
  NULL,
  xlim = c(1, 4), ylim = c(0, 1),
  xlab = "Length of Alligator", ylab = "Predictted Probability"
)
food_col <- c(F = 2, I = 3, O = 5)

sapply(c("F", "I", "O"), function(food) {
  lines(new_length_x$Length, prob_food[, food], col = food_col[food])
})

legend(1.75, 0.95, c("Fish", "Invertebrate", "Other"), lty = 1, col = 2:5)
```



是否相信来世

```
library(VGAM)
library(tidyr)
library(cdabookdb)
data("afterlife2")
ftable(afterlife2)
```

```
##              Believe Yes Undecided  No
## Race  Gender
## White Female         371        49  74
##       Male           250        45  71
## Black Female          64         9  15
##       Male            25         5  13
```

```
afterlife2_df <- spread(as.data.frame(afterlife2), Believe, Freq)
afterlife2.fit1 <- vglm(
  cbind(Yes, Undecided, No) ~ (Gender == "Female") + (Race == "White"),
  data = afterlife2_df, family = multinomial()
)

summary(afterlife2.fit1)
```

```
##
## Call:
## vglm(formula = cbind(Yes, Undecided, No) ~ (Gender == "Female") +
##     (Race == "White"), family = multinomial(), data = afterlife2_df)
##
##
## Pearson residuals:
##   log(mu[,1]/mu[,3]) log(mu[,2]/mu[,3])
## 1             -0.219             -0.114
## 2              0.228              0.111
## 3              0.471              0.230
## 4             -0.618             -0.280
##
## Coefficients:
##                           Estimate Std. Error z value
## (Intercept):1                0.883      0.243    3.64
## (Intercept):2               -0.758      0.361   -2.10
## Gender == "Female"TRUE:1     0.419      0.171    2.44
## Gender == "Female"TRUE:2     0.105      0.247    0.43
## Race == "White"TRUE:1        0.342      0.237    1.44
## Race == "White"TRUE:2        0.271      0.354    0.77
##                           Pr(>|z|)
## (Intercept):1              0.00027 ***
## (Intercept):2              0.03593 *
## Gender == "Female"TRUE:1   0.01452 *
## Gender == "Female"TRUE:2   0.66996
## Race == "White"TRUE:1      0.14934
## Race == "White"TRUE:2      0.44416
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of linear predictors:  2
##
## Names of linear predictors:
## log(mu[,1]/mu[,3]), log(mu[,2]/mu[,3])
##
## Residual deviance: 0.854 on 2 degrees of freedom
##
## Log-likelihood: -19.73 on 2 degrees of freedom
##
## Number of iterations: 3
##
```

```
## No Hauck-Donner effect found in any of the estimates
##
## Reference group is level  3  of the response
```

```
fitted(afterlife2.fit1)
```

```
##      Yes Undecided     No
## 1 0.7546   0.09956 0.1459
## 2 0.6783   0.12245 0.1993
## 3 0.7074   0.10018 0.1925
## 4 0.6222   0.12056 0.2573
```

6.2 有序响应变量的累积 logit 模型

政治意识形态和隶属党派的关系

```
library(VGAM)
library(tidyr)
library(cdabookdb)
data("ideology")
ftable(ideology)
```

```
##              Ideology VLib SLib Mod SCon VCon
## Gender Party
## Female Dem              44   47 118   23   32
##        Rep              18   28  86   39   48
## Male   Dem              36   34  53   18   23
##        Rep              12   18  62   45   51
```

```
ideology_df <- spread(as.data.frame(ideology), Ideology, Freq)

ide_m <- vglm(
  cbind(VLib, SLib, Mod, SCon, VCon) ~ Party == "Dem",
  data = ideology_df,
  family = cumulative(parallel = TRUE) # 累积概率且解释变量系数相同
)
summary(ide_m)
```

```
##
## Call:
## vglm(formula = cbind(VLib, SLib, Mod, SCon, VCon) ~ Party ==
```

```
##     "Dem", family = cumulative(parallel = TRUE), data = ideology_df)
##
##
## Pearson residuals:
##   logit(P[Y<=1]) logit(P[Y<=2]) logit(P[Y<=3])
## 1        -0.4630         -1.272          1.506
## 2        -0.0773          0.759          0.914
## 3         1.0080          1.339         -0.605
## 4        -0.4888         -0.489         -2.064
##   logit(P[Y<=4])
## 1         -0.681
## 2          0.918
## 3         -1.074
## 4          0.271
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept):1       -2.4690     0.1318  -18.73  < 2e-16 ***
## (Intercept):2       -1.4745     0.1091  -13.52  < 2e-16 ***
## (Intercept):3        0.2371     0.0948    2.50    0.012 *
## (Intercept):4        1.0695     0.1046   10.23  < 2e-16 ***
## Party == "Dem"TRUE   0.9745     0.1291    7.55  4.3e-14 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of linear predictors:  4
##
## Names of linear predictors:
## logit(P[Y<=1]), logit(P[Y<=2]), logit(P[Y<=3]), logit(P[Y<=4])
##
## Residual deviance: 15.9 on 11 degrees of freedom
##
## Log-likelihood: -47.84 on 11 degrees of freedom
##
## Number of iterations: 4
##
## No Hauck-Donner effect found in any of the estimates
##
## Exponentiated coefficients:
## Party == "Dem"TRUE
##               2.65
```

对心理健康建模

```
library(VGAM)
library(cdabookdb)
data("impairment")


impairment_m <- vglm(
  Impairment ~ SES + LifeEvents,
  family = cumulative(parallel = TRUE), # 累积概率且解释变量系数相同
  data = impairment
)
```

```
## Warning in eval(slot(family, "initialize")): response should
## be ordinal---see ordered()
```

```
summary(impairment_m)
```

```
##
## Call:
## vglm(formula = Impairment ~ SES + LifeEvents, family = cumulative(parallel = TRUE),
##     data = impairment)
##
##
## Pearson residuals:
##                      Min     1Q Median    3Q  Max
## logit(P[Y<=1]) -1.57 -0.705 -0.210 0.807 2.71
## logit(P[Y<=2]) -2.33 -0.467  0.266 0.690 1.61
## logit(P[Y<=3]) -3.69  0.120  0.204 0.419 1.89
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept):1   -0.282      0.623   -0.45   0.6510
## (Intercept):2    1.213      0.651    1.86   0.0625 .
## (Intercept):3    2.209      0.717    3.08   0.0021 **
## SES              1.111      0.614    1.81   0.0704 .
## LifeEvents      -0.319      0.119   -2.67   0.0076 **
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of linear predictors:  3
```

```
##
## Names of linear predictors:
## logit(P[Y<=1]), logit(P[Y<=2]), logit(P[Y<=3])
##
## Residual deviance: 99.1 on 115 degrees of freedom
##
## Log-likelihood: -49.55 on 115 degrees of freedom
##
## Number of iterations: 5
##
## No Hauck-Donner effect found in any of the estimates
##
## Exponentiated coefficients:
##        SES LifeEvents
##      3.038      0.727
```

6.3 成对类别有序 logit

再访政治意识形态

```
library(VGAM)
library(tidyr)
library(cdabookdb)
data("ideology")
ftable(ideology)
```

```
##              Ideology VLib SLib Mod SCon VCon
## Gender Party
## Female Dem              44   47 118   23   32
##        Rep              18   28  86   39   48
## Male   Dem              36   34  53   18   23
##        Rep              12   18  62   45   51
```

```
ideology_df <- spread(as.data.frame(ideology), Ideology, Freq)

ide_m <- vglm(
  cbind(VLib, SLib, Mod, SCon, VCon) ~ Party == "Dem",
  data = ideology_df,
  # 相邻类别 logit，更高且系数相同
  family = acat(reverse = TRUE, parallel = TRUE)
```

```
)
```

```
## Warning in vglm.fitter(x = x, y = y, w = w, offset = offset,
## Xm2 = Xm2, : some quantities such as z, residuals, SEs may
## be inaccurate due to convergence at a half-step
```

```
summary(ide_m)
```

```
##
## Call:
## vglm(formula = cbind(VLib, SLib, Mod, SCon, VCon) ~ Party ==
##     "Dem", family = acat(reverse = TRUE, parallel = TRUE), data = ideology_df)
##
##
## Pearson residuals:
##   loge(P[Y=1]/P[Y=2]) loge(P[Y=2]/P[Y=3])
## 1              -0.595              -1.117
## 2               0.125               0.463
## 3               0.714               1.505
## 4              -0.106              -0.620
##   loge(P[Y=3]/P[Y=4]) loge(P[Y=4]/P[Y=5])
## 1               1.730              -0.948
## 2               0.833               1.057
## 3              -0.525              -1.190
## 4              -2.247               0.554
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept):1         -0.439      0.140   -3.14   0.0017 **
## (Intercept):2         -1.172      0.112  -10.46  < 2e-16 ***
## (Intercept):3          0.732      0.109    6.72  1.8e-11 ***
## (Intercept):4         -0.368      0.121   -3.03   0.0025 **
## Party == "Dem"TRUE     0.435      0.060    7.25  4.1e-13 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of linear predictors:  4
##
## Names of linear predictors:
## loge(P[Y=1]/P[Y=2]), loge(P[Y=2]/P[Y=3]), loge(P[Y=3]/P[Y=4]), loge(P[Y=4]/P[Y=5])
##
```

```
## Residual deviance: 17.73 on 11 degrees of freedom
##
## Log-likelihood: -48.75 on 11 degrees of freedom
##
## Number of iterations: 4
##
## No Hauck-Donner effect found in any of the estimates
```

发育毒性研究

```
library(VGAM)
library(tidyr)
library(cdabookdb)
data("toxicity")

concentration <- as.numeric(rownames(toxicity))
toxicity.fit.cratio <- vglm(
  unclass(toxicity) ~ concentration,
  family=cratio(reverse = FALSE, parallel = FALSE)
)
summary(toxicity.fit.cratio)
```

```
##
## Call:
## vglm(formula = unclass(toxicity) ~ concentration, family = cratio(reverse = FALSE,
##     parallel = FALSE))
##
##
## Pearson residuals:
##      logit(P[Y>1|Y>=1]) logit(P[Y>2|Y>=2])
## 0                -1.190             -0.063
## 62.5             -1.060              1.480
## 125               0.586              0.446
## 250               1.596             -0.879
## 500              -0.629              0.858
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept):1    3.247934   0.157660    20.6   <2e-16 ***
## (Intercept):2    5.701902   0.330652    17.2   <2e-16 ***
## concentration:1 -0.006389   0.000435   -14.7   <2e-16 ***
```

```
## concentration:2 -0.017375   0.001213   -14.3   <2e-16 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of linear predictors:  2
##
## Names of linear predictors:
## logit(P[Y>1|Y>=1]), logit(P[Y>2|Y>=2])
##
## Residual deviance: 11.84 on 6 degrees of freedom
##
## Log-likelihood: -26.35 on 6 degrees of freedom
##
## Number of iterations: 5
##
## Warning: Hauck-Donner effect detected in the following estimate(s):
## '(Intercept):1', 'concentration:2'
```

此处结课本符号相反：课本计算的是 $\text{logit}(P[Y=1|Y>=1])$ 和 $\text{logit}(P[Y=2|Y>=2])$，这里计算的是 $\text{logit}(P[Y>1|Y>=1])$ 和 $\text{logit}(P[Y>2|Y>=2])$

6.4 条件独立性检验

工作满意度和收入

```
library(cdabookdb)
data("job_satisfaction2")
```

```
gender <- factor(
  rep(dimnames(job_satisfaction2)$Gender, each = 4),
  dimnames(job_satisfaction2)$Gender
)
income <- rep(c(3, 10, 20, 35), times = 2)
```

首先拟合两个累积 logit 模型（考虑收入效应和不考虑收入效应）

```
# 有收入效应的模型
job2.fit1 <- vglm(
  as.matrix(ftable(job_satisfaction2)) ~ gender + income,
```

```
  family = cumulative(parallel = TRUE)
)
summary(job2.fit1)
```

```
##
## Call:
## vglm(formula = as.matrix(ftable(job_satisfaction2)) ~ gender +
##     income, family = cumulative(parallel = TRUE))
##
##
## Pearson residuals:
##                     Min     1Q  Median    3Q   Max
## logit(P[Y<=1]) -0.858 -0.588 -0.4348 0.201 1.270
## logit(P[Y<=2]) -1.200 -0.457 -0.1883 0.652 2.044
## logit(P[Y<=3]) -1.008 -0.371  0.0964 0.396 0.589
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept):1  -2.5795     0.5618   -4.59  4.4e-06 ***
## (Intercept):2  -0.8939     0.3603   -2.48    0.013 *
## (Intercept):3   2.0781     0.4206    4.94  7.8e-07 ***
## genderMale     -0.0257     0.4274   -0.06    0.952
## income         -0.0444     0.0185   -2.40    0.017 *
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of linear predictors:  3
##
## Names of linear predictors:
## logit(P[Y<=1]), logit(P[Y<=2]), logit(P[Y<=3])
##
## Residual deviance: 13.95 on 19 degrees of freedom
##
## Log-likelihood: -28.06 on 19 degrees of freedom
##
## Number of iterations: 5
##
## No Hauck-Donner effect found in any of the estimates
##
## Exponentiated coefficients:
## genderMale     income
```

```
##     0.9747     0.9565
```

```
# 没有收入效应的模型
job2.fit2 <- vglm(
  as.matrix(ftable(job_satisfaction2)) ~ gender,
  family = cumulative(parallel = TRUE)
)
summary(job2.fit2)
```

```
##
## Call:
## vglm(formula = as.matrix(ftable(job_satisfaction2)) ~ gender,
##     family = cumulative(parallel = TRUE))
##
##
## Pearson residuals:
##                       Min     1Q   Median    3Q   Max
## logit(P[Y<=1]) -0.842 -0.655 -0.48385 0.403 2.105
## logit(P[Y<=2]) -1.208 -0.698 -0.02072 0.796 1.961
## logit(P[Y<=3]) -1.424 -0.553 -0.00333 0.780 0.924
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept):1   -3.078      0.526   -5.85  4.9e-09 ***
## (Intercept):2   -1.420      0.293   -4.85  1.2e-06 ***
## (Intercept):3    1.426      0.293    4.87  1.1e-06 ***
## genderMale      -0.412      0.402   -1.02     0.31
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of linear predictors:  3
##
## Names of linear predictors:
## logit(P[Y<=1]), logit(P[Y<=2]), logit(P[Y<=3])
##
## Residual deviance: 19.62 on 20 degrees of freedom
##
## Log-likelihood: -30.89 on 20 degrees of freedom
##
## Number of iterations: 5
##
## Warning: Hauck-Donner effect detected in the following estimate(s):
```

```
## '(Intercept):1'
##
## Exponentiated coefficients:
## genderMale
##     0.6626
```

接着对比两个模型

```
# 对比两个模型
c(deviance = deviance(job2.fit1), df = df.residual(job2.fit1))
```

```
## deviance       df
##    13.95    19.00
```

```
c(deviance = deviance(job2.fit2), df = df.residual(job2.fit2))
```

```
## deviance       df
##    19.62    20.00
```

```
df_diff <- df.residual(job2.fit2) - df.residual(job2.fit1)
deviance_diff <- deviance(job2.fit2) - deviance(job2.fit1)
1 - pchisq(deviance_diff, df_diff)
```

```
## [1] 0.01725
```

接着拟合两个基线-类别 logit 模型，再进行对比（这次收入是因子而不是数值）

```
income_factor <- factor(income)
# 有收入效应的模型
job2.fit3 <- vglm(
  as.matrix(ftable(job_satisfaction2)) ~ gender + income_factor,
  family = multinomial()
)
summary(job2.fit3)
```

```
##
## Call:
## vglm(formula = as.matrix(ftable(job_satisfaction2)) ~ gender +
##     income_factor, family = multinomial())
##
##
## Pearson residuals:
##                     log(mu[,1]/mu[,4]) log(mu[,2]/mu[,4])
```

```
## Female_<5000                -3.50e-01             0.065
## Female_5000-15000            4.15e-01            -0.687
## Female_15000-25000          -1.36e-05             0.474
## Female_>25000                1.62e-05             1.069
## Male_<5000                   6.52e-01            -0.106
## Male_5000-15000             -6.86e-01             1.121
## Male_15000-25000             1.44e-05            -0.564
## Male_>25000                 -1.03e-05            -0.748
##                     log(mu[,3]/mu[,4])
## Female_<5000                     0.403
## Female_5000-15000                0.128
## Female_15000-25000              -0.507
## Female_>25000                   -0.194
## Male_<5000                      -0.731
## Male_5000-15000                 -0.221
## Male_15000-25000                 0.590
## Male_>25000                      0.145
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept):1        -0.3785     0.9613   -0.39    0.694
## (Intercept):2         0.3108     0.7859    0.40    0.692
## (Intercept):3         1.5077     0.6539    2.31    0.021 *
## genderMale:1         -0.1122     1.2827   -0.09    0.930
## genderMale:2         -0.0956     0.7676   -0.12    0.901
## genderMale:3         -0.1761     0.5331   -0.33    0.741
## income_factor10:1    -0.2832     1.2594   -0.22    0.822
## income_factor10:2     0.1216     1.0005    0.12    0.903
## income_factor10:3     0.2454     0.8406    0.29    0.770
## income_factor20:1   -19.3686  4263.9687      NA       NA
## income_factor20:2    -2.3489     1.3149   -1.79    0.074 .
## income_factor20:3    -0.8046     0.7825   -1.03    0.304
## income_factor35:1   -19.2924  4161.9671      NA       NA
## income_factor35:2    -1.2266     1.0740   -1.14    0.253
## income_factor35:3    -0.9040     0.8160   -1.11    0.268
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of linear predictors:  3
##
## Names of linear predictors:
## log(mu[,1]/mu[,4]), log(mu[,2]/mu[,4]), log(mu[,3]/mu[,4])
```

```
##
## Residual deviance: 7.093 on 9 degrees of freedom
##
## Log-likelihood: -24.63 on 9 degrees of freedom
##
## Number of iterations: 17
##
## Warning: Hauck-Donner effect detected in the following estimate(s):
## 'income_factor20:1', 'income_factor35:1'
##
## Reference group is level  4  of the response
```

```
# 没有收入效应的模型
job2.fit4 <- vglm(
  as.matrix(ftable(job_satisfaction2)) ~ gender,
  family = multinomial()
)
summary(job2.fit4)
```

```
##
## Call:
## vglm(formula = as.matrix(ftable(job_satisfaction2)) ~ gender,
##     family = multinomial())
##
##
## Pearson residuals:
##                       Min     1Q  Median    3Q   Max
## log(mu[,1]/mu[,4]) -1.169 -0.738 -0.4024 0.536 2.482
## log(mu[,2]/mu[,4]) -1.145 -0.948  0.2135 0.620 2.021
## log(mu[,3]/mu[,4]) -0.818 -0.618 -0.0343 0.430 0.679
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept):1   -1.386      0.645   -2.15  0.03174 *
## (Intercept):2   -0.288      0.441   -0.65  0.51414
## (Intercept):3    1.204      0.329    3.66  0.00025 ***
## genderMale:1    -1.012      1.228   -0.82  0.41000
## genderMale:2    -0.501      0.697   -0.72  0.47226
## genderMale:3    -0.466      0.493   -0.95  0.34383
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Number of linear predictors:  3
##
## Names of linear predictors:
## log(mu[,1]/mu[,4]), log(mu[,2]/mu[,4]), log(mu[,3]/mu[,4])
##
## Residual deviance: 19.37 on 18 degrees of freedom
##
## Log-likelihood: -30.76 on 18 degrees of freedom
##
## Number of iterations: 5
##
## No Hauck-Donner effect found in any of the estimates
##
## Reference group is level  4  of the response
```

```
# 对比两个模型
c(deviance = deviance(job2.fit3), df = df.residual(job2.fit3))
```

```
## deviance       df
##    7.093    9.000
```

```
c(deviance = deviance(job2.fit4), df = df.residual(job2.fit4))
```

```
## deviance       df
##    19.37    18.00
```

```
df_diff <- df.residual(job2.fit4) - df.residual(job2.fit3)
deviance_diff <- deviance(job2.fit4) - deviance(job2.fit3)
1 - pchisq(deviance_diff, df_diff)
```

```
## [1] 0.1983
```

课后题

附录 A　配套 R 包使用介绍

A.1　安装

把已经编译好的 R 包解压到 R 的 library 目录（可在 R 中运行.libPaths() 查看）下

此外，这份代码文档中用到包已经被列为 cdabookdb 的建议包。要安装这些建议包可以在安装时指定参数

可以这样获取建议包并安装

```
suggested_pkgs <- packageDescription("cdabookdb")$Suggests
suggested_pkgs <- strsplit(suggested_pkgs, ",\\s*")[[1]]
suggested_pkgs
```

```
##  [1] "knitr"              "rmarkdown"
##  [3] "Fahrmeir"           "binom"
##  [5] "dplyr"              "MASS"
##  [7] "pROC"               "ResourceSelection"
##  [9] "ROCR"               "tidyr"
## [11] "VGAM"
```

```
# 如果没有安装则进行安装
lapply(suggested_pkgs, function(pkg) {
  if (system.file(package = pkg) == '') install.packages(pkg)
})
```

A.2　使用说明

可以使用 data(package = "cdabookdb") 查看包中包含的数据集及其说明

```
library(cdabookdb)
data(package = "cdabookdb")$results[, 3]
```

```
##  [1] "AIDS_treatment"           "AZT"
##  [3] "MBtest1"                  "MBtest2"
##  [5] "MBtest3"                  "UCBAdmissions"
##  [7] "UFAdmissions"             "accident_seatbelt1"
##  [9] "accident_seatbelt2"       "accident_seatbelt3"
## [11] "afterlife1"               "afterlife2"
## [13] "albumin"                  "alligators1"
## [15] "alligators2"              "aspirin"
## [17] "athlete_graduate"         "birth_control"
## [19] "blood_pressure"           "cancer_remission"
## [21] "chip_imperfection"        "cholesterol"
## [23] "credit_score"             "creditcard"
## [25] "deathpenalty1"            "deathpenalty2"
## [27] "edu_aspiration"           "environmental_protection"
## [29] "football_arrest"          "gender_party"
## [31] "government_spending"      "happiness1"
## [33] "happiness2"               "happiness3"
## [35] "horseshoecrabs"           "ideology"
## [37] "impairment"               "incontinent"
## [39] "job_satisfaction1"        "job_satisfaction2"
## [41] "job_satisfaction3"        "kyphosis_age"
## [43] "larynx_cancer"            "lungcancer_treatment"
## [45] "malformation"             "marijuana"
## [47] "marital_happiness"        "merit_pay_race"
## [49] "missing_persons"          "osteosarcoma"
## [51] "premarital_sex1"          "premarital_sex2"
## [53] "promotion_race"           "psych_diag_drugs"
## [55] "rabbit_penicillin"        "race_party"
## [57] "religious_belief"         "smoking_cd"
## [59] "smoking_lungcancer"       "smoking_lungcancer_cn"
## [61] "smoking_mi"               "snoring_heartdisease"
## [63] "teen_sex"                 "teenager_crime"
## [65] "temperature_distress"     "throat"
## [67] "toxicity"                 "traincollisions"
## [69] "treatment1"               "treatment2"
## [71] "treatment3"               "white_black_acceptance"
```

使用 `data(DATANAME)` 可以引入数据集。

此外，cdabookfunc 包中包含了几个有用的函数

```
sort(getNamespaceExports("cdabookfunc"))
```

```
## [1] "binom_inference"           "binom_mid_pvalue"
```

```
## [3] "dfbetas_logit_sas"         "find_data_by_title"
## [5] "find_data_by_var"          "independent_test_of_table"
## [7] "influence_logit_sas"       "oddsratio"
## [9] "samplesize_prop"
```

函数的用处如下表所示

函数名	说明	参考章节
find_data_by_title	根据数据的 title 查找数据集	无
find_data_by_var	根据数据中包含的变量名查找数据集	无
binom_inference	二项分布的推断	1.4.2 节
binom_mid_pvalue	二项分布中点 P 值	1.4.5 节
oddsratio	计算优势比	2.3 节
independent_test_of_table	三种列联表的独立性检验方法	2.4-2.5 节
dfbetas_logit_sas	用 SAS 的方法计算 logistic 回归的 dfbetas	5.2.7 节
influence_logit_sas	用 SAS 的方法进行 logistic 回归的诊断	5.2.7 节
samplesize_prop	计算比较两个比例时所需的样本量	5.5.1 节

前两个函数是用于从 cdabookdb（默认，也可以是指定包或者全部已安装的包等，具体可查看函数的帮助信息）的一大堆数据集中寻找所需的数据集。从第三个函数开始是用于方便实现书上的代码结果。

附录 B　教材数据列表

B.1　正文案例数据

以下表格为教材正文的案例用到的案例数据集（均可在 `cdabookdb` 中找到）

章节	案例名称	数据集
2.1	关于来世	afterlife1
2.2	阿司匹林与心脏病（列联表检验）	aspirin
2.3	阿司匹林与心脏病（优势比）	aspirin
2.3	吸烟状态与心肌梗死	smoking_mi
2.4	性别和党派认同	gender_party
2.5	饮酒与婴儿畸形	malformation
2.6	小样本的精确推断	无
2.7	死刑判决案例	deathpenalty1
2.7	临床试验	treatment1
3.2	打鼾与心脏病	snoring_heartdisease
3.3	母鲎及其追随者（泊松 GLM）	horseshoecrabs
3.3	母鲎及其追随者（负二项 GLM）	horseshoecrabs
3.3	英国的火车事故	traincollisions
3.4	打鼾与心脏病	snoring_heartdisease
4.1	母鲎及其追随者（logistic 回归）	horseshoecrabs
4.3	AZT 和 AIDS	AZT
4.4	母鲎及其追随者（多元 logistic）	horseshoecrabs
5.1	母鲎及其追随者（模型选择）	horseshoecrabs
5.1	母鲎及其追随者（预测功效）	horseshoecrabs
5.2	母鲎及其追随者（模型 LR 检验）	horseshoecrabs
5.2	AZT 和 AIDS（拟合优度）	AZT
5.2	母鲎及其追随者（HM 检验）	horseshoecrabs
5.2	佛罗里达大学研究生入学	UFAdmissions
5.2	心脏病与血压的关系	blood_pressure
5.3	稀疏数据的临床试验结果	treatment3
5.4	晋升能力	promotion_race
5.5	样本量计算	无
6.1	钝吻鳄食物选择	alligators1

6.1	是否相信来世	afterlife2
6.2	政治意识形态和隶属党派的关系	ideology
6.2	对心理健康建模	impairment
6.3	再访政治意识形态	ideology
6.3	发育毒性研究	toxicity
6.4	工作满意度和收入	job_satisfaction2

B.2 习题数据

以下表格为教材的习题用到的数据集（均可在 `cdabookdb` 中找到）

习题	数据集
2.16	smoking_lungcancer
2.18	happiness1
2.19	race_party
2.21	teenager_crime
2.22	psych_diag_drugs
2.23	religious_belief
2.27	edu_aspiration
2.3	larynx_cancer
2.33	deathpenalty2
3.3	malformation
3.4	malformation
3.5	snoring_heartdisease
3.6	snoring_heartdisease
3.7	horseshoecrabs
3.8	horseshoecrabs
3.9	creditcard
3.1	cancer_remission
3.11	chip_imperfection
3.12	chip_imperfection
3.13	horseshoecrabs
3.14	horseshoecrabs
3.18	football_arrest
3.19	traincollisions
3.2	smoking_cd
4.1	cancer_remission
4.2	cancer_remission
4.4	snoring_heartdisease
4.5	temperature_distress
4.6	creditcard

4.7	kyphosis_age
4.8	horseshoecrabs
4.12	deathpenalty2
4.13	deathpenalty2
4.14	AZT
4.15	merit_pay_race
4.16	MBtest
4.17	MBtest
4.2	treatment2
4.22	horseshoecrabs
4.24	throat
4.25	horseshoecrabs
4.26	horseshoecrabs
4.27	horseshoecrabs
4.29	teen_sex
4.3	athlete_graduate
4.31	marijuana
4.32	albumin
4.33	job_satisfaction_survey
4.37	deathpenalty1
5.1	horseshoecrabs
5.2	horseshoecrabs
5.3	horseshoecrabs
5.4	MBtest1
5.6	MBtest1
5.7	MBtest2
5.9	cancer_remission
5.1	horseshoecrabs
5.11	horseshoecrabs
5.12	premarital_sex1
5.13	credit_score
5.15	missing_persons
5.17	deathpenalty1
5.18	smoking_lungcancer_cn
5.19	UCBAdmissions
5.2	malformation
5.21	malformation
5.23	rabbit_penicillin
5.24	rabbit_penicillin
5.25	osteosarcoma
5.26	incontinent
5.29	horseshoecrabs
6.2	alligators1

6.3	alligators2
6.4	afterlife2
6.6	marital_happiness
6.7	marital_happiness
6.8	lungcancer_treatment
6.1	impairment
6.11	job_satisfaction2
6.12	happiness2
6.13	job_satisfaction2
6.14	afterlife2
6.15	job_satisfaction2
6.16	cholesterol
6.17	accident_seatbelt1
6.19	job_satisfaction3
6.21	happiness3
7.1	afterlife1
7.2	afterlife1
7.3	white_black_acceptance
7.4	AIDS_treatment
7.5	deathpenalty1
7.6	MBtest3
7.7	MBtest3
7.8	MBtest3
7.9	UCBAdmissions
7.1	accident_seatbelt3
7.12	accident_seatbelt2
7.13	government_spending
7.14	premarital_sex1
7.15	marijuana
7.16	accident_seatbelt2
7.21	government_spending
7.22	marijuana
7.24	birth_control