cda10 月 29 日课堂

林振炜

2019 年 10 月 29 日

0.1 1 snoring and heart disease

- question1

```
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
```

```
scores.a <- c(0,2,4,5)
scores.b <- c(0,2,4,6)
scores.c <- 0:3
scores.d <- 1:4

snoring_logit <- glm(
  snoring_heartdisease ~ scores.a, family = binomial(link = 'logit' )
)
snoring_logit
```

```
##
## Call:  glm(formula = snoring_heartdisease ~ scores.a, family = binomial(link = "logit")
##
```

```
## Coefficients:
## (Intercept)     scores.a
##     -3.8662       0.3973
##
## Degrees of Freedom: 3 Total (i.e. Null);  2 Residual
## Null Deviance:       65.9
## Residual Deviance: 2.809     AIC: 27.06
```

```
snoring_logit <- glm(
  snoring_heartdisease ~ scores.b, family = binomial(link = 'logit' )
)
snoring_logit
```

```
##
## Call:  glm(formula = snoring_heartdisease ~ scores.b, family = binomial(link = "logit")
##
## Coefficients:
## (Intercept)     scores.b
##     -3.7774       0.3273
##
## Degrees of Freedom: 3 Total (i.e. Null);  2 Residual
## Null Deviance:       65.9
## Residual Deviance: 6.24  AIC: 30.49
```

```
snoring_linger <- glm(
  snoring_heartdisease ~ scores.c, family = binomial(link = 'logit' )
)
snoring_logit
```

```
##
## Call:  glm(formula = snoring_heartdisease ~ scores.b, family = binomial(link = "logit")
##
## Coefficients:
## (Intercept)     scores.b
##     -3.7774       0.3273
##
## Degrees of Freedom: 3 Total (i.e. Null);  2 Residual
```

```
## Null Deviance:       65.9
## Residual Deviance: 6.24  AIC: 30.49
```

```
snoring_logit <- glm(
  snoring_heartdisease ~ scores.d, family = binomial(link = 'logit' )
)
snoring_logit
```

```
##
## Call:  glm(formula = snoring_heartdisease ~ scores.d, family = binomial(link = "logit")
##
## Coefficients:
## (Intercept)     scores.d
##     -4.4319       0.6545
##
## Degrees of Freedom: 3 Total (i.e. Null);  2 Residual
## Null Deviance:       65.9
## Residual Deviance: 6.24  AIC: 30.49
```

- question2

```
snoring_linear <- glm(
snoring_heartdisease ~ scores.a, family = binomial(link = "identity")
)
snoring_logistics <- glm(
snoring_heartdisease ~ scores.a, family = binomial(link = "logit")
)
snoring_probit <- glm(
snoring_heartdisease ~ scores.a, family = binomial(link = "probit")
)
model_list <- list(snoring_linear, snoring_logistics, snoring_probit)

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.a     0.020  0.397  0.188
```

```r
pred_prob <- sapply(model_list, predict, type = "response")
colnames(pred_prob) <- c("linear", "logit", "probit")
round(pred_prob, digits = 3)
```

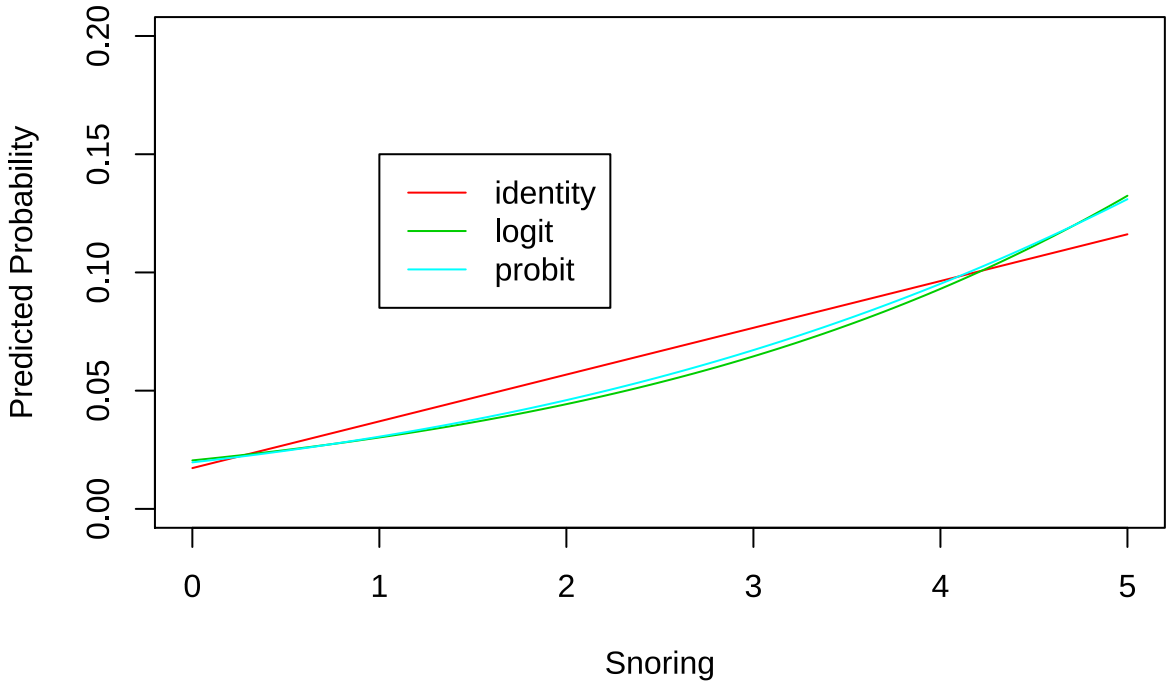
```
##                     linear logit probit
## Never                0.017 0.021  0.020
## Occasional           0.057 0.044  0.046
## Nearly every night   0.096 0.093  0.095
## Every night          0.116 0.132  0.131
```

```r
snoring_new <- seq(0,5,0.01)
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, data.frame(scores.a = snoring_new), type = "response");pred_resu
lines(
snoring_new, pred_result, type = "l",
lty = 1, col = line_col[m$family$link]
)
}
)
```

```
## [[1]]
## NULL
##
## [[2]]
## NULL
##
## [[3]]
## NULL
```

```
legend(1, 0.15, names(line_col), col = line_col, lty = 1)
```



2 - q1

```
data('horseshoecrabs')
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
## -2.8526  -1.9884  -0.4933   1.0970   4.9221
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.30476    0.54224  -6.095  1.1e-09 ***
## Width        0.16405    0.01997   8.216  < 2e-16 ***
## ---
```

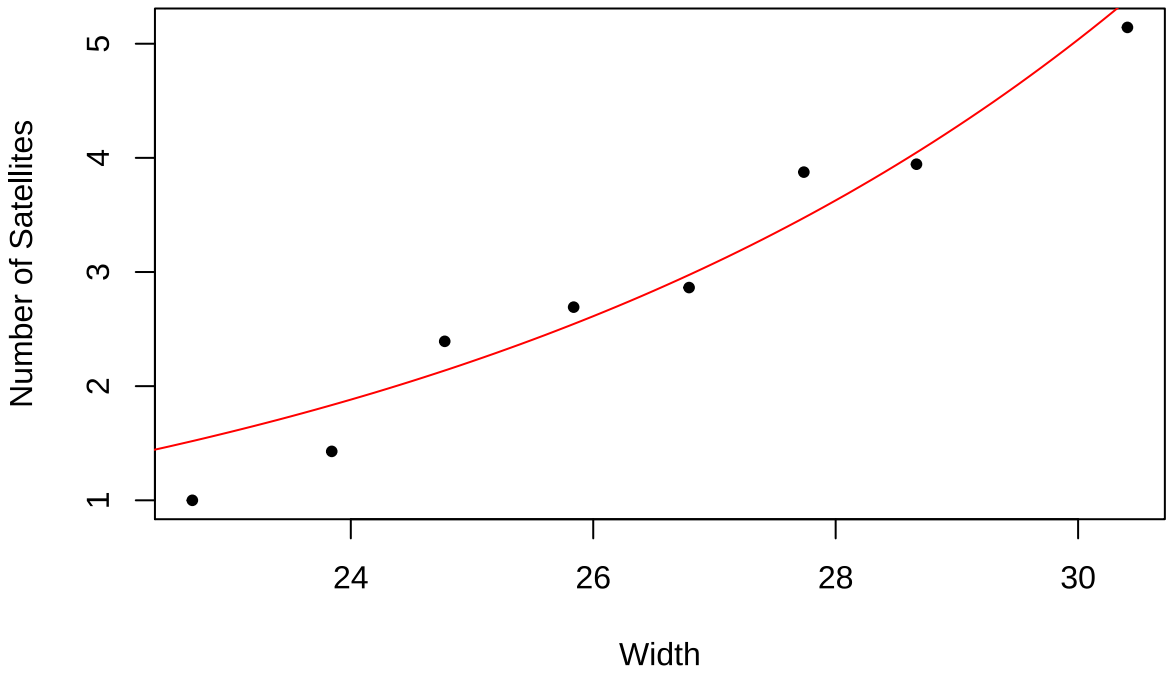
```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 632.79  on 172  degrees of freedom
## Residual deviance: 567.88  on 171  degrees of freedom
## AIC: 927.18
##
## Number of Fisher Scoring iterations: 6
```

```r
#group by width first, calculate the average width and number of satellites
width_group <- cut(horseshoecrabs$Width,
                   breaks = c(0, 23.25 + 0:6, Inf),
                   dig.lab = 4)
mean_width_vs_sat <- sapply(levels(width_group),
                            function(x){
                              # Declare that is grouped by width_group
                              sub <- subset(horseshoecrabs,width_group==x)
                              c(mean(sub$Satellites),mean(sub$Width))
}
)

mean_satellite <- mean_width_vs_sat[1,] # average satellite of Each group
mean_width <- mean_width_vs_sat[2,] # average width of each group

plot(
mean_satellite ~ mean_width,
pch = 20, # set type of points as solid ball
xlab = "Width", ylab = "Number of Satellites" # labels of axes
)
x <- seq(22, 32, 0.1)
y_m1 <- predict(m1, data.frame(Width = x), type = "response")

lines(x, y_m1, type = "l", col = 2)
```




```
# grouped data with poisson loglinear model
po_log_m <- glm(
  mean_satellite ~ mean_width,
  family = poisson(link = 'log'),
  data = data.frame(mean_satellite,mean_width),
  start = coef(m1)
);summary(po_log_m)
```

```
##
## Call:
## glm(formula = mean_satellite ~ mean_width, family = poisson(link = "log"),
##     data = data.frame(mean_satellite, mean_width), start = coef(m1))
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -0.32971  -0.17962  -0.02783   0.19032   0.27335
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept) -3.80578    2.41755  -1.574   0.1154
## mean_width   0.18148    0.08794   2.064   0.0391 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 4.74059  on 7  degrees of freedom
## Residual deviance: 0.34557  on 6  degrees of freedom
## AIC: Inf
##
## Number of Fisher Scoring iterations: 3
```

```r
#fit the grouped/ungrouped data with Negative loglinear model
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
## -1.7821  -1.4124  -0.2507   0.4780   2.0218
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4.05144    1.07767  -3.759 0.000234 ***
## Width        0.19203    0.04053   4.738 4.52e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for Negative Binomial(0.9091) family taken to be 0.8495001)
##
##     Null deviance: 213.63  on 172  degrees of freedom
## Residual deviance: 196.33  on 171  degrees of freedom
## AIC: 755.29
##
## Number of Fisher Scoring iterations: 5
```

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

grouped_m <- glm.nb(mean_satellite~mean_width,data = data.frame(mean_satellite,mean_width)
```

```
##
## Call:
## glm.nb(formula = mean_satellite ~ mean_width, data = data.frame(mean_satellite,
##     mean_width), init.theta = 809199.3517, link = log)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -0.32971  -0.17962  -0.02783   0.19032   0.27335
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.80578    2.41756  -1.574   0.1154
## mean_width   0.18148    0.08794   2.064   0.0391 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(809199.4) family taken to be 1)
##
##     Null deviance: 4.74057  on 7  degrees of freedom
## Residual deviance: 0.34557  on 6  degrees of freedom
## AIC: 29.268
##
## Number of Fisher Scoring iterations: 1
##
##
```

```
##                Theta:  809199
##            Std. Err.:  83480518
## Warning while fitting theta: iteration limit reached
##
##  2 x log-likelihood:  -23.268
```

0.2 3

- q1

```
data('traincollisions')
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
## -2.0580  -0.7825  -0.0826   0.3775   3.3873
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.21142    0.15892  -26.50  < 2e-16 ***
## year0       -0.03292    0.01076   -3.06  0.00222 **
## ---
## 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: 133.52
##
## Number of Fisher Scoring iterations: 5
```

```
m_nb <- glm(
  TrRd~year0,data = traincollisions,offset = log(traincollisions$KM)
);summary(m_nb)
```

```
##
## Call:
## glm(formula = TrRd ~ year0, data = traincollisions, offset = log(traincollisions$KM))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.8673  -1.6823  -0.0354   0.7589   8.4934
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.21039    0.97446  -0.216   0.8307
## year0       -0.11899    0.05975  -1.992   0.0566 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 7.246704)
##
##     Null deviance: 224.40  on 28  degrees of freedom
## Residual deviance: 195.66  on 27  degrees of freedom
## AIC: 143.66
##
## Number of Fisher Scoring iterations: 2
```