Curso de Estadística básica para Data Scientists

Dae-Jin Lee < lee.daejin@gmail.com >

TEMA 7. Regresión para datos binarios y de conteo

Índice

Regresar a la página principal

1. Logistic regression

A logistic regression is typically used when there is one dichotomous outcome variable (such as winning or losing), and a continuous predictor variable which is related to the probability or odds of the outcome variable. It can also be used with categorical predictors, and with multiple predictors.

1.1. ESR and Plasma Proteins

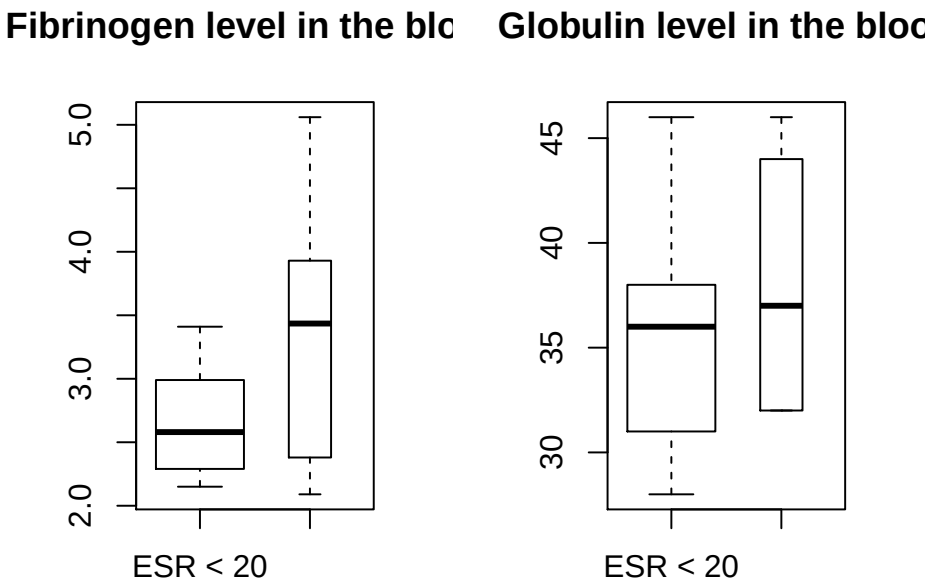
The erythrocyte sedimentation rate (ESR) is the rate at which red blood cells (erythrocytes) settle out of suspension in blood plasma, when measured under standard conditions. If the ESR increases when the level of certain proteins in the blood plasma rise in association with conditions such as rheumatic diseases, chronic infections and malignant diseases, its determination might be useful in screening blood samples taken from people suspected of suffering from one of the conditions mentioned. The absolute value of the ESR is not of great importance; rather, less than 20mm/hr indicates a "healthy" individual. To assess whether the ESR is a useful diagnostic tool, the question of interest is whether there is any association between the probability of an ESR reading greater than 20mm/hr and the levels of the two plasma proteins. If there is not then the determination of ESR would not be useful for diagnostic purposes. A data frame with 32 observations on the following 3 variables.

- `fibrinogen` the fibrinogen level in the blood.
- `globulin` the globulin level in the blood.
- `ESR` the erythrocyte sedimentation rate, either less or greater 20 mm / hour.

```
data("plasma", package = "HSAUR")
head(plasma)
```

```
##   fibrinogen globulin      ESR
## 1       2.52       38 ESR < 20
## 2       2.56       31 ESR < 20
## 3       2.19       33 ESR < 20
## 4       2.18       31 ESR < 20
## 5       3.41       37 ESR < 20
## 6       2.46       36 ESR < 20
```

```r
layout(matrix(1:2, ncol = 2))
boxplot(fibrinogen ~ ESR, data = plasma, varwidth = TRUE, main="Fibrinogen level in the bloc
boxplot(globulin ~ ESR, data = plasma, varwidth = TRUE, main="Globulin level in the blood")
```



The question of interest is whether there is any association between the probability of an ESR reading greater than 20mm/hr and the levels of the two plasma proteins. If there is not then the determination of ESR would not be useful for diagnostic purposes.

Since the response variable is binary, a multiple regression model is not suitable for a regression analysis.

We may write

$$\Pr(y_i = 1) = \pi_i \qquad \Pr(y_i = 0) = 1 - \pi_i$$

Normally, we will have a set of covariates $X = (x_1, ..., x_p)$ associated with each individual, and our goal will be to investigate the relationship between the response probability $\pi = \pi(X)$ and the explanatory variables.

So instead of modelling the expected value of the response directly as a linear function of explanatory variables, a suitable transformation is modelled. In this case the most suitable transformation is the logistic or logit function of π leading to the model

$$\text{logit}(\pi) = \text{logit}\left(\frac{\pi}{1-\pi}\right) = \beta_0 + \beta_1 x_1 + ... + \beta_p x_p$$

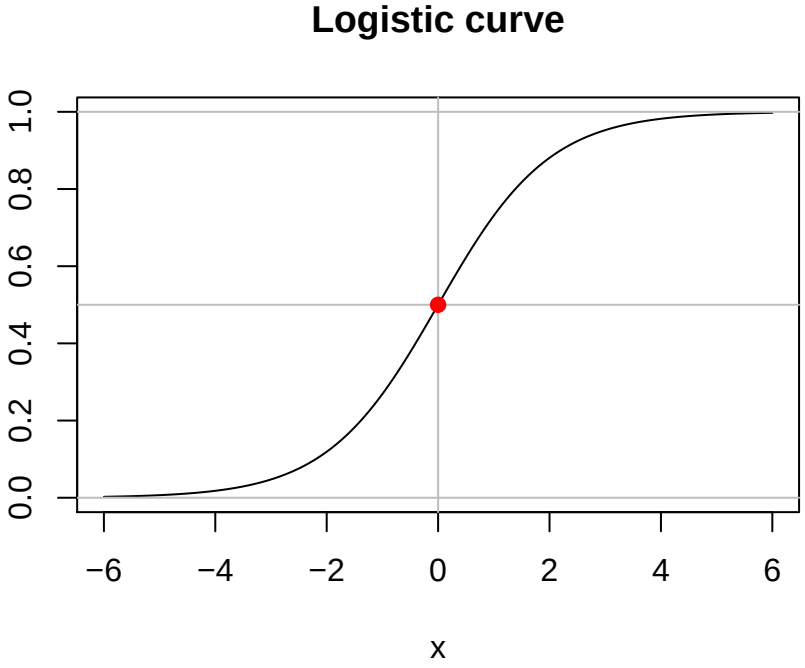
The logit of a probability is simply the log of the odds of the response taking the value one or logit transformation, of p: $logit(p) = \log(p/1-p)$. Logit is sometimes called "log odds." Because of the properties of odds given in the list above, the logit has these properties:

- If `odds(y=1s) = 1`, then `logit(p) = 0`.
- If `odds(y=1) < 1`, then `logit(p) < 0`.
- If `odds(y=1) > 1`, then `logit(p) > 0`.

The logit transform fails if p = 0‘.

When the response is a binary (dichotomous) variable, and x is numeric, logistic regression fits a logistic curve to the relationship between x and y. Hence, logistic regression is linear regression on the logit transform of y, where y is the proportion (or probability) of success at each value of x. However, you should avoid the temptation to do a traditional least-squares regression at this point, as neither the normality nor the homoscedasticity assumption will be met.

```
x <- seq(-6,6,0.01)
logistic <- exp(x)/(1+exp(x))
plot(x,logistic,t='l',main="Logistic curve",ylab="")
abline(h=c(0,0.5,1),v=0,col="grey")
points(0,0.5,pch=19,col=2)
```



Logistic regression model in `R` can be fitted using the function `glm`. First, we start with a model that includes only a single predictor `fibrinogen`

```
plasma_glm_1 <- glm(ESR ~ fibrinogen, data = plasma,family = binomial())
summary(plasma_glm_1)
```

```
##
## Call:
## glm(formula = ESR ~ fibrinogen, family = binomial(), data = plasma)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9298  -0.5399  -0.4382  -0.3356   2.4794
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -6.8451     2.7703  -2.471   0.0135 *
## fibrinogen    1.8271     0.9009   2.028   0.0425 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 30.885  on 31  degrees of freedom
## Residual deviance: 24.840  on 30  degrees of freedom
## AIC: 28.84
##
## Number of Fisher Scoring iterations: 5
```

We see that the regression coefficient for `fibrinogen` is significant at the 5 % level. An increase of one unit in this variable increases the log-odds in favour of an ESR value greater than 20 by an estimated 1,83 with 95 % confidence interval.

```
confint(plasma_glm_1,parm="fibrinogen")
```

```
## Waiting for profiling to be done...
```

```
##     2.5 %    97.5 %
## 0.3387619 3.9984921
```

These values are more helpful if converted to the corresponding values for the odds themselves by exponentiating the estimate

```
exp(coef(plasma_glm_1)["fibrinogen"])
```

```
## fibrinogen
##   6.215715
```

and the confidence interval

```
exp(confint(plasma_glm_1, parm = "fibrinogen"))
```

```
## Waiting for profiling to be done...
```

```
##     2.5 %    97.5 %
##  1.403209 54.515884
```

The confidence interval is very wide because there are few observations overall and very few where the ESR value is greater than 20. Nevertheless it seems likely that increased values of fibrinogen lead to a greater probability of an ESR value greater than 20. We can now fit a logistic regression model that includes both explanatory variables using the code

```
plasma_glm_2 <- glm(ESR ~ fibrinogen + globulin, data = plasma,family = binomial())
summary(plasma_glm_2)
```

```
##
## Call:
## glm(formula = ESR ~ fibrinogen + globulin, family = binomial(),
##     data = plasma)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9683  -0.6122  -0.3458  -0.2116   2.2636
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -12.7921     5.7963  -2.207   0.0273 *
## fibrinogen    1.9104     0.9710   1.967   0.0491 *
## globulin      0.1558     0.1195   1.303   0.1925
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 30.885  on 31  degrees of freedom
```

```
## Residual deviance: 22.971  on 29  degrees of freedom
## AIC: 28.971
##
## Number of Fisher Scoring iterations: 5
```

The coefficient for gamma globulin is not significantly different from zero.

Both nested models can be compared using a likelihood ratio test with `anova` function

```
anova(plasma_glm_1, plasma_glm_2, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: ESR ~ fibrinogen
## Model 2: ESR ~ fibrinogen + globulin
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        30     24.840
## 2        29     22.971  1   1.8692   0.1716
```

So we conclude that gamma globulin is not associated with ESR level.

The plot clearly shows the increasing probability of an ESR value above 20 (larger circles) as the values of fibrinogen, and to a lesser extent, gamma globulin, increase.

```
prob <- predict(plasma_glm_2,type="response")
plot(globulin ~ fibrinogen, data = plasma, xlim = c(2, 6),ylim = c(25, 55), pch = ".")
symbols(plasma$fibrinogen, plasma$globulin, circles = prob,add = TRUE)
```