

Logistic Regression Ch5

F.A. Barrios Instituto de Neurobiología UNAM

2020-11-23

```
library(tidyverse)
library(here)
library(wesanderson)
library(rstatix)
library(HSAUR2)
library(car)
library(multcomp)

setwd("~/Dropbox/GitHub/Class2020")
wcgs <- read_csv("DataRegressBook/Chap2/wcgs.csv")
```

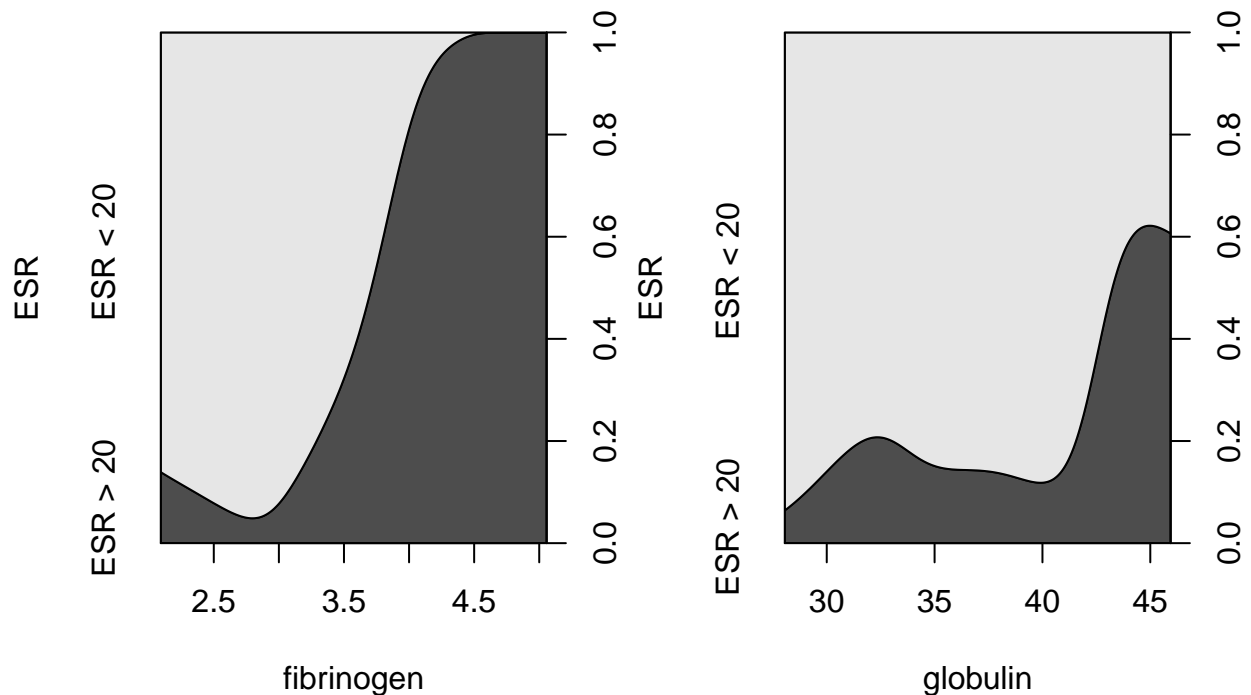
Logistic Regression

Example from HSAUR (Chapter 7, in HSAUR3)

Introduction

The erythrocyte sedimentation rate (ESR) is the rate at which red blood cells (erythrocytes) settle out of suspension in the 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, Collett and Jemain (1985) collected the data in HSAUR2. 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.

```
# Using plasma data from HSAUR
data("plasma", package = "HSAUR2")
layout(matrix(1:2, ncol = 2))
# cdpplot computes and plots conditional densities describing how the conditional distribution of a cate
cdplot(ESR ~ fibrinogen, data = plasma)
cdplot(ESR ~ globulin, data = plasma)
```



To estimate a logistic regression model in R the glm (General Linear Model) is used, for binomial distribution the glm() function default to a logistic model.

```
# glm general linear model default is logistic for binomial distribution
```

```
plasma_glm01 <- glm(ESR ~ fibrinogen, data = plasma, family = binomial())
summary(plasma_glm01)
```

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

From these results we see that the regression coefficients for fibrinogen is significant at the 5% level. An increase of one unit in this variable increases the log-odds on favor of an ESR value greater then 20 by estimated 1.83 with 95% confidence interval:

```
# coeff fibrinogen is significant 5%
# one unit change in this variable increases the log-odds in favor of ESR > 20mm/hr by 1.83
confint(plasma_glm01, parm = "fibrinogen")
```

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

```
exp(coef(plasma_glm01)["fibrinogen"])
```

```
fibrinogen
6.215715
```

```
exp(confint(plasma_glm01, parm = "fibrinogen"))
```

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

These are the values of the odds themselves (by exponentiating the estimate). So **increased values of fibrinogen lead to a greater probability of an ESR value greater than 20.**

```
# full model with two variables
plasma_glm02 <- glm(ESR ~ fibrinogen + globulin, data = plasma, family = binomial())
summary(plasma_glm02)
```

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

Comparing the residual deviance of the models: residual deviance 01: 24.84 residual deviance 02: 22.971 -> 1.869 (1.87), to test for significance R take the lgm with a χ^2 the 1.87 we conclude that **the globulin has no influence in the ESR**. To compare the two nested models (with fibrinogen and fibrinogen + gamma globulin) we can estimate the ANOVA of the models (Pr of 0.1716)

```
anova(plasma_glm01, plasma_glm02, test = "Chisq")
```

Analysis of Deviance Table

```

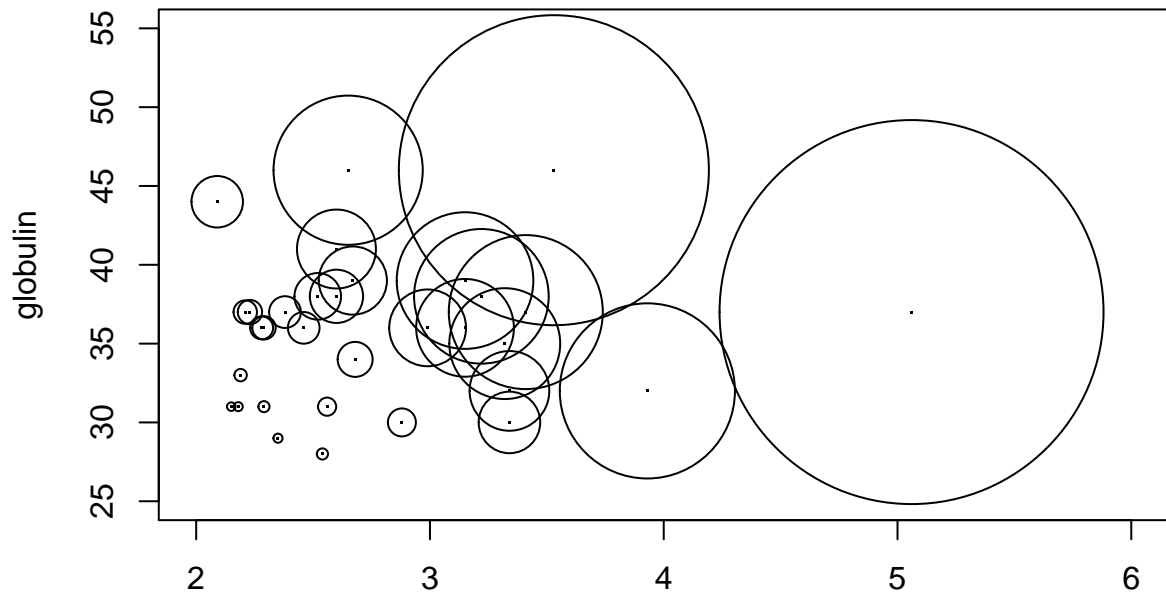
Model 1: ESR ~ fibrinogen
Model 2: ESR ~ fibrinogen + globulin
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1      30      24.840
2      29      22.971  1    1.8692   0.1716

# Estimates conditional probability of a ESR > 20 for all observations

prob <- predict(plasma_glm02, type = "response")
layout(matrix(1:1, ncol = 1))

plot(globulin ~ fibrinogen, data = plasma, xlim = c(2, 6), ylim = c(25, 55), pch = ".")
symbols(plasma$fibrinogen, plasma$globulin, circles = prob, add = TRUE)

```



fibrinogen

##

Single predictor models

In the linear model the “expected value” of the $Y|x$ is estimated with a simple linear model

$$E[y|x] = \beta_0 + \beta_1 x$$

And to model a function that has a binary outcome, in particular if we assign the expected value $E[y|x]$ to the observed proportion ($P(x)$) of a variable given x that take the form of a simple linear model in x

$$P(x) = E[y|x] = \beta_0 + \beta_1 x$$

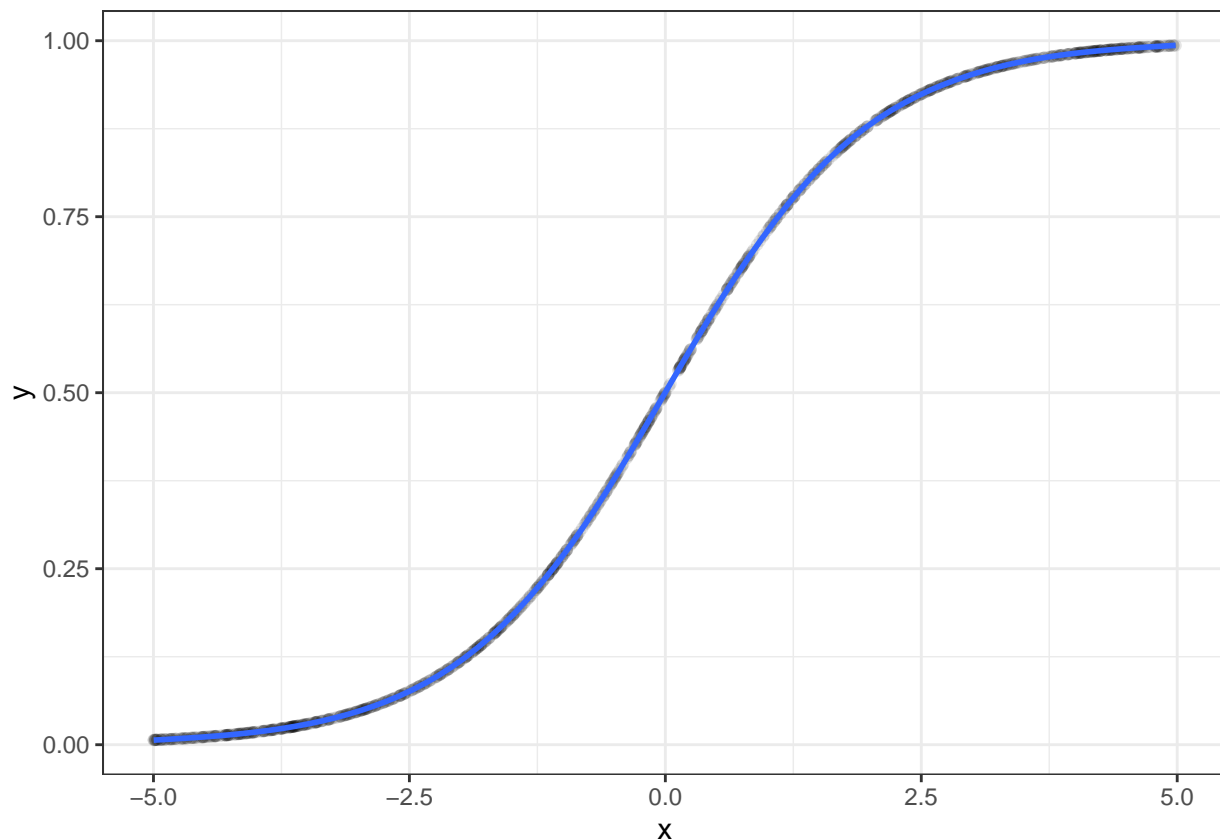
At this equation if x is a binary predictor taking on the values 0 or 1, $P(0) = \beta_0 + \beta_1(0)$ and $P(1) = \beta_0 + \beta_1(1)$, then $P(1) - P(0) = \beta_1$ therefore the effect of increasing x one unit is to add an increment β_1 to the outcome. This is the risk difference associated with a unit increase in x . Models with this property are often referred to as *additive risk models*.

The statistical machinery which allowed us to use this linear model to make inferences about the strength of relationship in expected value $E[y|x]$ linear model, required that the outcome variable follow an approximate normal distribution. For binary outcome, this assumption is incorrect. And, the outcome in the model that we are studying now represents a probability or risk. Thus, any estimates of the regression coefficients must constrain the estimated probability to lie between zero and one for the model to make sense. $P(x) = E[y|x]$.

The smooth S-shaped curve in the following plot is known as the logistic model. The exponential model is also known as log linear because it specifies that the logarithm of the outcome risk is linear in x

```
# generate fake data
set.seed(42)
n <- 1000
x <- (runif(n) * 10) - 5
y <- (exp(x) / (1 + exp(x)))
df <- data.frame(x = x, y = y)

# Logistic plot
p1 <- ggplot(df, aes(x, y)) +
  geom_point(alpha = 0.1) +
  geom_smooth(se = FALSE) +
  theme_bw()
p1
```



The logistic model allows for a smooth change in risk throughout the range of x , and has the property that risk increases slowly up to a “threshold” range of x . The mathematical model is the following for a simple linear model

$$P(x) = \frac{e^{(\beta_0 + \beta_1 x)}}{1 + e^{(\beta_0 + \beta_1 x)}}$$

in terms of the odds of the outcome associated with the predictor x , the model can also be expressed as

$$\frac{P(x)}{1 - P(x)} = e^{(\beta_0 + \beta_1 x)}$$

If x takes the values 0 or 1 (binary predictor), the ratio of the odds for these two values are

$$\frac{\frac{P(1)}{1-P(1)}}{\frac{P(0)}{1-P(0)}} = \frac{e^{(\beta_0+\beta_1)}}{e^{\beta_0}} = e^{\beta_1}$$

Therefore the logit model is a “multiplicative risk model”. The most widely used model for binary outcomes in clinical and epidemiological applications, and forms the basis of logistic regression modeling. The random part of the model specifies the distribution of the outcome variable y_i , conditional on the observed value x_i of the predictor (where the subscript i denotes the value for a particular subject). For binary outcomes, this distribution is called the binomial distribution and is completely specified by the mean of y_i conditional on the value x_i . To summarize, the logistic model makes the following assumptions about the outcome y_i :

1. y_i follows a Binomial distribution
2. The mean $E[y|x] = P(x)$ is given by the logistic function
3. Values of the outcome are statistically independent

Interpretation of Regression Coefficients

So, the estimated logistic-regression model is given by

$$\log\left[\frac{\hat{\mu}(x)}{1-\hat{\mu}(x)}\right] = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k$$

If exponentiate both sides of the equation, we get

$$\frac{\hat{\mu}(x)}{1-\hat{\mu}(x)} = \exp(\beta_0) \times \exp(\beta_1 x_1) \times \exp(\beta_2 x_2) \times \cdots \times \exp(\beta_k x_k)$$

where the left hand side of the equation, $\frac{\hat{\mu}(x)}{1-\hat{\mu}(x)}$, gives the *fitted odds* of success, **the fitted probability of success divided by the fitted probability of failure**. Exponentiating the model removes the logarithms and changes the model in the log-odds scale to one that is multiplicative, in this log odds scale.

For the WCGS data and the variable Coronary Heart Disease (CHD) and age, the β_1 is the age slope of the fitted logistic model. The outcome of the model is the log odds of CHD risk and the relationship with age, the slope coefficient β_1 gives the change in the log odds of chd69 associated with the model.

```
wcgs <- mutate(wcgs, chd69 = factor(chd69))
# For table 5.2
CHD_glm01 <- glm(chd69 ~ age, data = wcgs, family = binomial())
S(CHD_glm01)
```

```
Call: glm(formula = chd69 ~ age, family = binomial(), data = wcgs)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-5.93952	0.54932	-10.813	< 2e-16 ***
age	0.07442	0.01130	6.585	4.56e-11 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

```
Null deviance: 1781.2 on 3153 degrees of freedom
Residual deviance: 1738.4 on 3152 degrees of freedom
```

logLik	df	AIC	BIC
-869.18	2	1742.36	1754.47

Number of Fisher Scoring iterations: 5

Exponentiated Coefficients and Confidence Bounds

	Estimate	2.5 %	97.5 %
(Intercept)	0.002633304	0.0008898193	0.007676359
age	1.077261913	1.0536638869	1.101432899

```
#confint(CHD_glm01, parm = "age")  
# To estimate the model  
exp(coef(CHD_glm01)["age"])
```

```
age  
1.077262
```

The link transformation is the exponentiation, to obtain the odds.