Logistic Regression

The ASTA team

Contents

1 Introduction to logistic regression

1.1 Binary response

- We consider a binary response y with outcome 1 or 0. This might be a code indicating whether a person is able or unable to perform a given task.
- Furthermore, we are given an explanatory variable x, which is numeric, e.g. age.
- We shall study models for
$$P(y = 1 \,|\, x)$$
i.e. the probability that a person of age x is able to complete the task.
- We shall see methods for determining whether or not age actually influences the probability, i.e. is y independent of x?

1.2 A linear model

$$P(y = 1 \,|\, x) = \alpha + \beta x$$

is simple, but often inappropiate. If β is positive and x sufficiently large, then the probability exceeds 1.

2 Simple logistic regression

2.1 Logistic model

Instead we consider the **odds** that the person is able to complete the task

$$\texttt{Odds}(y=1\,|\,x)=\frac{P(y=1\,|\,x)}{P(y=0\,|\,x)}=\frac{P(y=1\,|\,x)}{1-P(y=1\,|\,x)}$$

which can have any positive value.

The logistic model is defined as:

$$\texttt{logit}(P(y=1\,|\,x))=\log(\texttt{Odds}(y=1\,|\,x))=\alpha+\beta x$$

The function `logit`$(p)=\log(\frac{p}{1-p})$ - i.e. **log of odds** - is termed **the logistic transformation**.

Remark that log odds can be any number, where zero corresponds to $P(y=1\,|\,x)=0.5$. Solving $\alpha+\beta x=0$ shows that at age $x_0=-\alpha/\beta$ you have fifty-fifty chance of solving the task.

2.2 Logistic transformation

- The function `logit()` (remember to load `mosaic` first) can be used to calculate the logistic transformation:

```
p <- seq(0.1, 0.9, by = 0.2)
p
```

```
## [1] 0.1 0.3 0.5 0.7 0.9
```

```
l <- logit(p)
l
```

```
## [1] -2.1972246 -0.8472979  0.0000000  0.8472979  2.1972246
```

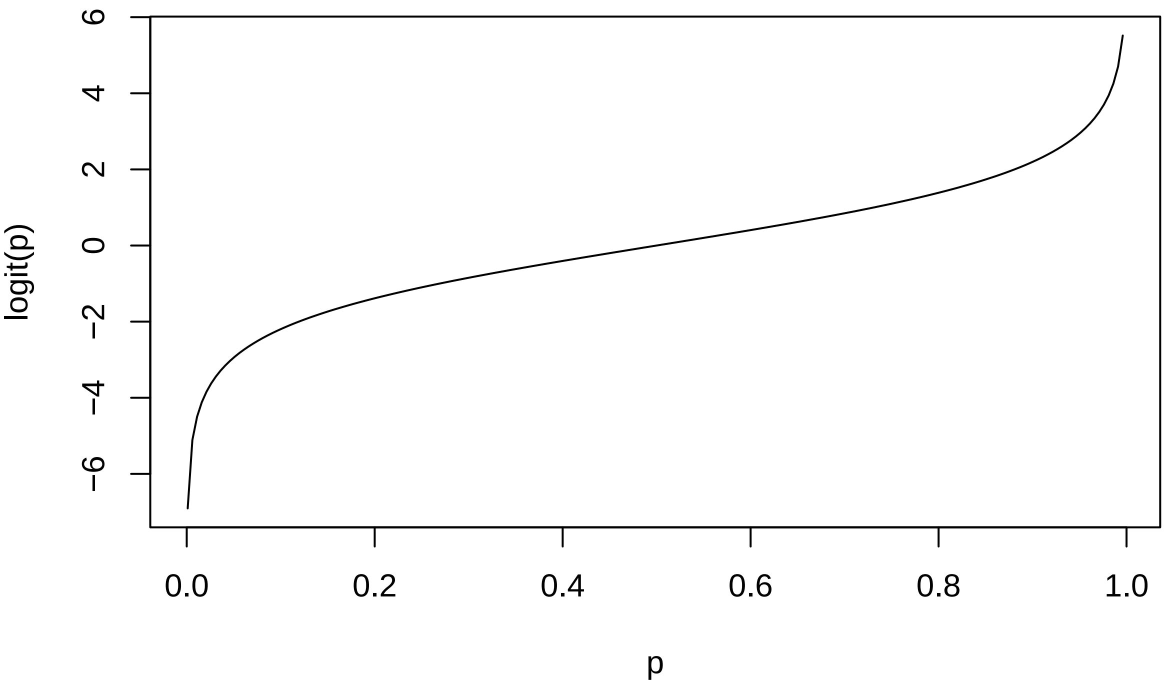
- The inverse logistic transformation `ilogit()` applied to the transformed values can recover the original probabilities:

```
ilogit(l)
```

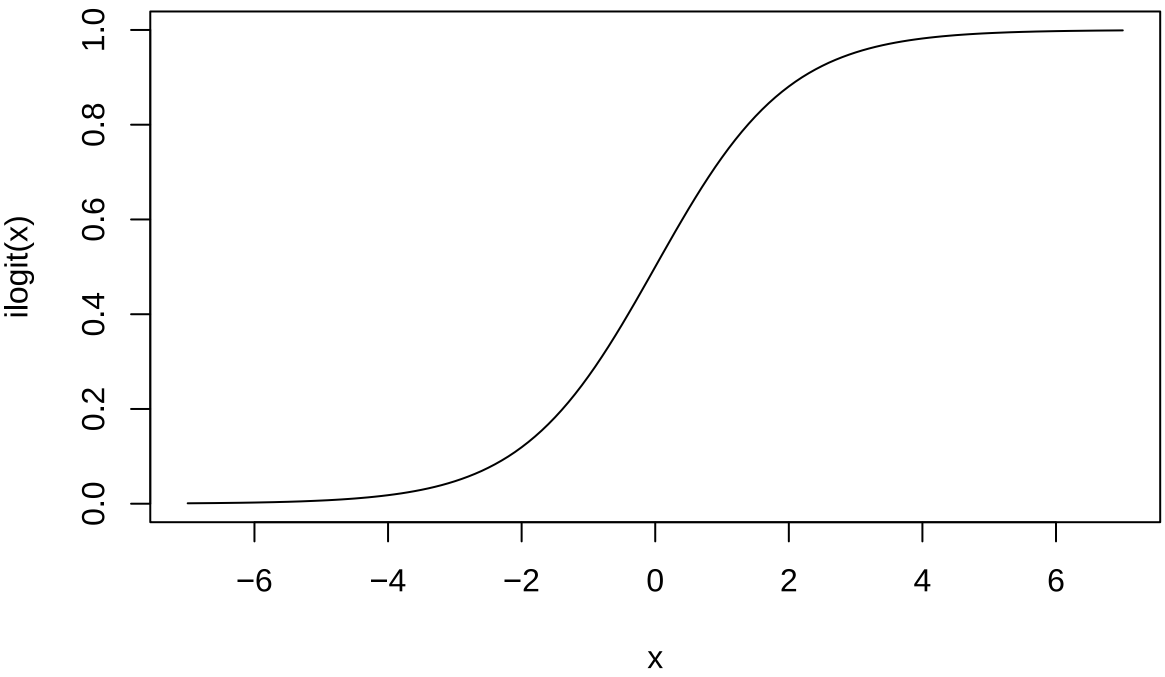
```
## [1] 0.1 0.3 0.5 0.7 0.9
```

Plot of logistic function and inverse logistic

```
p=seq(0.001,0.999,by=0.005)
plot(p,logit(p),type="l")
```



```
x=seq(-7,7,by= 0.1)
plot(x,ilogit(x),type="l")
```



2.3 Odds-ratio

Interpretation of β:

What happens to odds, if we increase age by 1 year?

Consider the so-called **odds-ratio**:

$$\frac{\texttt{Odds}(y=1\,|\,x+1)}{\texttt{Odds}(y=1\,|\,x)} = \frac{\exp(\alpha+\beta(x+1))}{\exp(\alpha+\beta x)} = \exp(\beta)$$

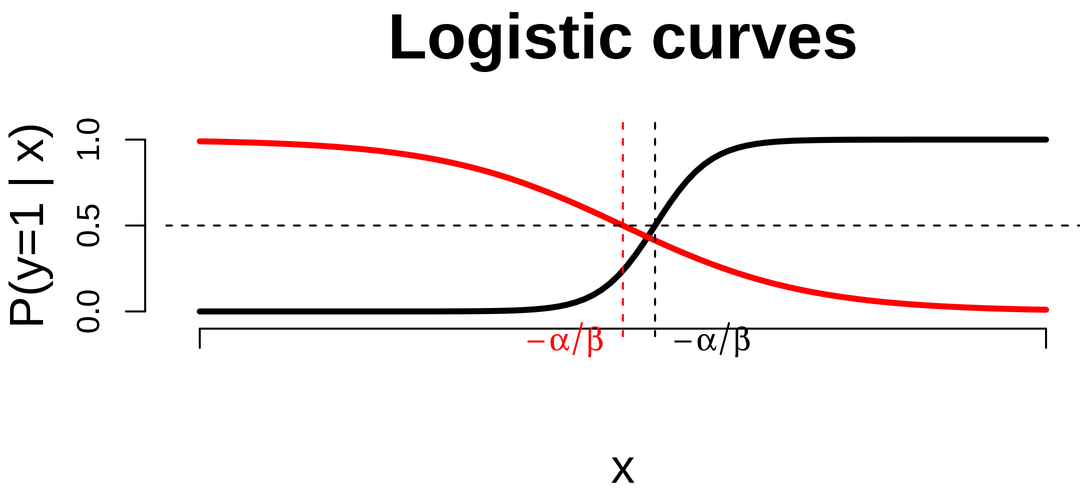
where we see, that $\exp(\beta)$ equals the odds for age $x+1$ relative to odds at age x.

This means that when age increase by 1 year, then the relative change

$$\frac{\exp(\alpha+\beta(x+1))-\exp(\alpha+\beta x)}{\exp(\alpha+\beta x)}$$

in odds is given by $100(\exp(\beta)-1)\%$.

2.4 Simple logistic regression



Examples of logistic curves for $P(y=1|x)$. The black curve has a positive β-value (=10), whereas the red has a negative β (=-3).

In addition we note that:

- Increasing the absolute value of β yields a steeper curve.
- When $P(y=1\,|\,x)=\frac{1}{2}$ then logit is zero, i.e. $\alpha+\beta x=0$.

This means that at age $x=-\frac{\alpha}{\beta}$ you have 50% chance to perform the task.

2.5 Example: Credit card data

We shall investigate if income is a good predictor of whether or not you have a credit card.

- Data structure: For each level of income, we let `n` denote the number of persons with that income, and `credit` how many of these that carries a credit card.

```
creInc <- read.csv("https://asta.math.aau.dk/datasets?file=income-credit.csv")
```

```
head(creInc)
```

```
##   Income  n credit
## 1     12  1      0
## 2     13  1      0
## 3     14  8      2
## 4     15 14      2
## 5     16  9      0
## 6     17  8      2
```

2.6 Example: Fitting the model

```
modelFit <- glm(cbind(credit,n-credit) ~ Income, data = creInc, family = binomial)
```

- `cbind` gives a matrix with two column vectors: `credit` and `n-credit`, where the latter is the vector counting the number of persons without a credit card.
- The response has the form `cbind(credit,n-credit)`.
- We need to use the function `glm` (generalized linear model).
- The argument `family=binomial` tells the function that the data has binomial variation. Leaving out this argument will lead `R` to believe that data follows a normal distribution - as with `lm`.
- The function `coef` extracts the coefficients (estimates of parameters) from the model summary:

```
coef(summary(modelFit))
```

```
##                Estimate Std. Error   z value     Pr(>|z|)
## (Intercept) -3.5179469 0.71033573 -4.952513 7.326117e-07
## Income       0.1054089 0.02615743  4.029788 5.582714e-05
```

2.7 Test of no effect

```
coef(summary(modelFit))
```

```
##                Estimate Std. Error   z value     Pr(>|z|)
## (Intercept) -3.5179469 0.71033573 -4.952513 7.326117e-07
## Income       0.1054089 0.02615743  4.029788 5.582714e-05
```

Our model for dependence of odds of having a credit card related to income(x) is

$$\texttt{logit}(x) = \alpha + \beta x$$

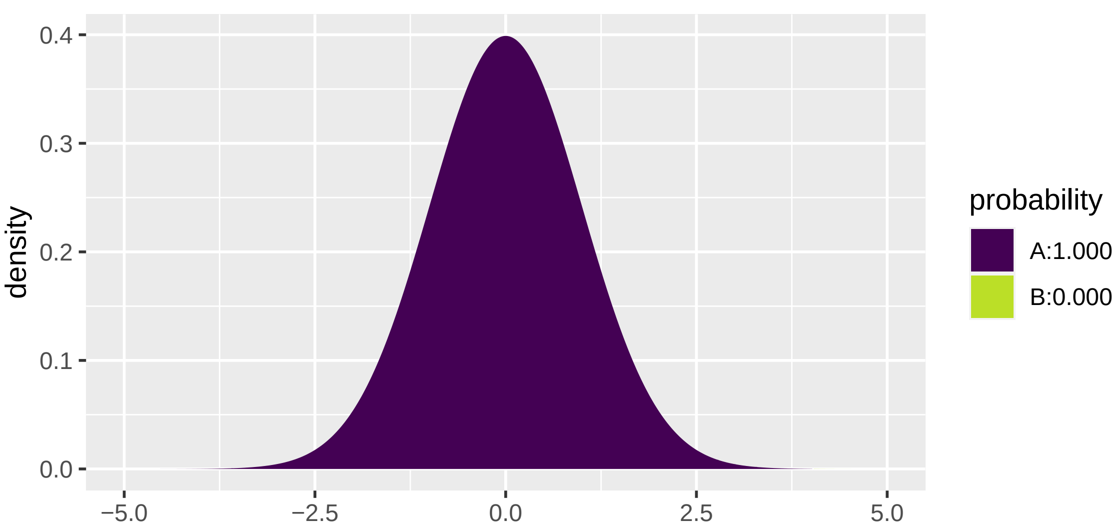
The hypothesis of no relation between income and ability to obtain a credit card corresponds to

$$H_0 : \quad \beta = 0$$

with the alternative $\beta \neq 0$. Inspecting the summary reveals that $\hat{\beta} = 0.1054$ is more than 4 standard errors away from zero.

With a z-score equal to 4.03 we get the tail probability

```
ptail <- 2*(1-pdist("norm",4.03,xlim=c(-5,5)))
```



```
ptail
```

```
## [1] 5.577685e-05
```

Which is very significant - as reflected by the p-value.

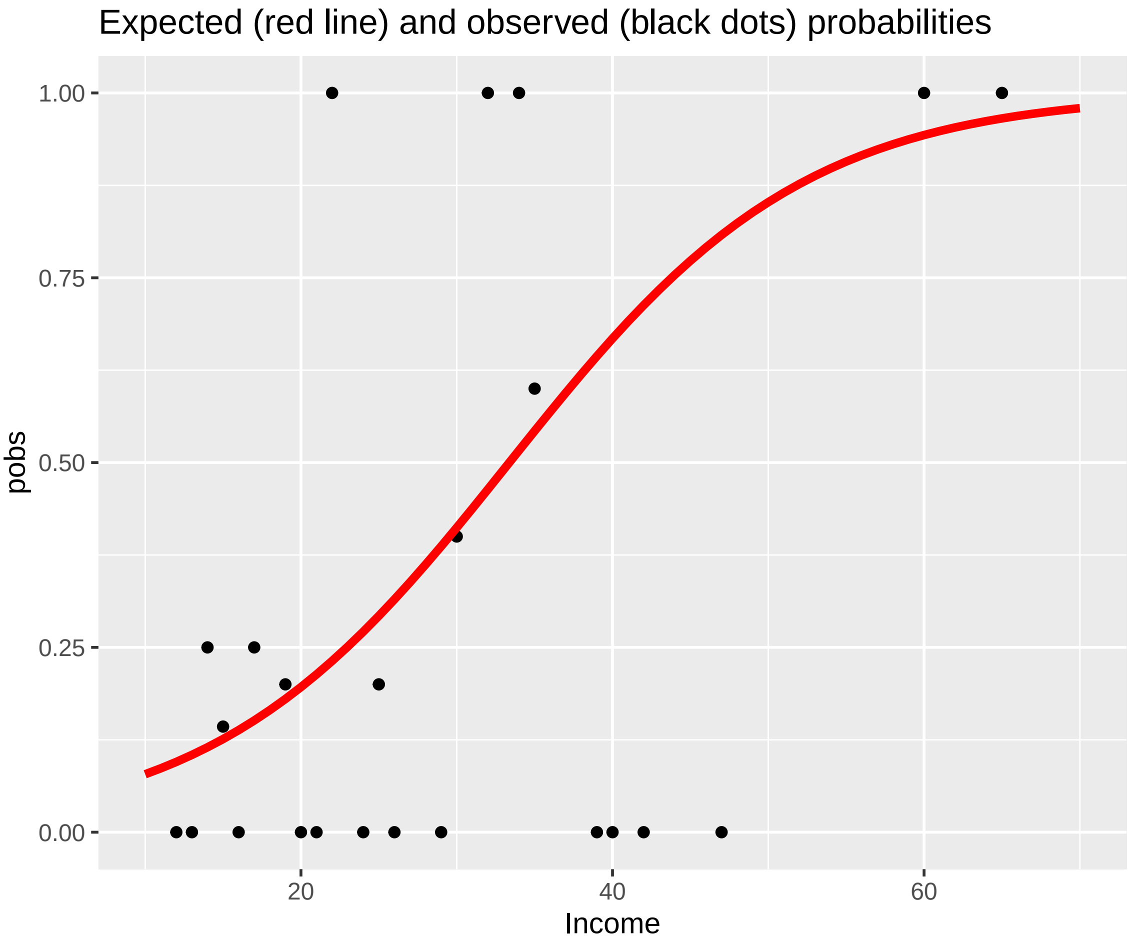
2.8 Confidence interval for odds ratio

From the summary:

- $\hat{\beta} = 0.10541$ and hence $\exp(\hat{\beta}) - 1 = 0.11$. If income increases by 1000 euro, then odds increases by 11%.
- Standard error on $\hat{\beta}$ is 0.02616 and hence a 95% confidence interval for log-odds ratio is $\hat{\beta} \pm 1.96 \times 0.02616 = (0.054; 0, 157)$.
- Corresponding interval for odds ratio: $\exp((0.054; 0, 157)) = (1.056; 1.170)$,

  i.e. the increase in odds is - with confidence 95% - between 5.6% and 17%.

2.9 Plot of model predictions against actual data



- Tendency is fairly clear and very significant.
- Due to low sample size at some income levels, the deviations are quite large.

3 Multiple logistic regression

3.1 Several numeric predictors

We generalize the model to the case, where we have k predictors $x_1, x_2, \dots, x_k$. Where some might be dummies for a factor.

$$\texttt{logit}(P(y = 1 \,|\, x_1, x_2, \dots, x_k)) = \alpha + \beta_1 x_1 + \cdots + \beta_k x_k$$

Interpretation of β-values is unaltered: If we fix $x_2, \dots, x_k$ and increase x_1 by one unit, then the relative change in odds is given by $\exp(\beta_1) - 1$.

3.2 Example

Wisconsin Breast Cancer Database covers 683 observations of 10 variables in relation to examining tumors in the breast.

- Nine clinical variables with a score between 0 and 10.
- The binary variable `Class` with levels `benign/malignant`.
- By default `R` orders the levels lexicografically and chooses the first level as reference ($y = 0$). Hence `benign` is reference, and we model odds of `malignant`.

We shall work with only 4 of the predictors, where two of these have been discretized.

```
BC <- read.table("https://asta.math.aau.dk/datasets?file=BC0.dat",header=TRUE)
head(BC)
```

```
##   nuclei cromatin Size.low Size.medium Shape.low     Class
## 1      1        3     TRUE       FALSE      TRUE    benign
## 2     10        3    FALSE        TRUE     FALSE    benign
## 3      2        3     TRUE       FALSE      TRUE    benign
## 4      4        3    FALSE       FALSE     FALSE    benign
## 5      1        3     TRUE       FALSE      TRUE    benign
## 6     10        9    FALSE       FALSE     FALSE malignant
```

3.3 Global test of no effects

First we fit the model `mainEffects` with main effect of all predictors - remember the notation `~ .` for all predictors. Then we fit the model `noEffects` with no predictors.

```
mainEffects <- glm(factor(Class)~., data=BC, family=binomial)
noEffects <- glm(factor(Class)~1, data=BC, family=binomial)
```

First we want to test, whether there is any effect of the predictors, i.e the null hypothesis

$$H_0: \quad \beta_1 = \beta_2 = \beta_3 = \beta_4 = \beta_5 = 0$$

3.4 Example

Similarly to `lm` we can use the function `anova` to compare `mainEffects` and `noEffects`. Only difference is that we need to tell the function that the test is a chi-square test and not an F-test.

```
anova(noEffects, mainEffects, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: factor(Class) ~ 1
## Model 2: factor(Class) ~ nuclei + cromatin + Size.low + Size.medium +
##     Shape.low
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1       682     884.35
## 2       677     135.06  5   749.29 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

`mainEffects` is a much better model.

The test statistic is the Deviance (749.29), which should be small.

It is evaluated in a chi-square with 5 (the number of parameters equal to zero under the nul hypothesis) degrees of freedom.

The 95%-critical value for the $\chi^2(5)$ distribution is 11.07 and the p-value is in practice zero.

3.5 Test of influence of a given predictor

```
round(coef(summary(mainEffects)),4)
```

```
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -0.7090     0.8570 -0.8274   0.4080
## nuclei             0.4403     0.0823  5.3484   0.0000
## cromatin           0.5058     0.1444  3.5026   0.0005
## Size.lowTRUE      -3.6154     0.8081 -4.4740   0.0000
## Size.mediumTRUE   -2.3773     0.7188 -3.3074   0.0009
## Shape.lowTRUE     -2.1490     0.6054 -3.5496   0.0004
```

For each predictor p can we test the hypothesis:

$$H_0 : \quad \beta_p = 0$$

- Looking at the `z`-values, there is a clear effect of all 5 predictors. Which of course is also supported by the p-values.

3.6 Prediction and classification

```
BC$pred <- round(predict(mainEffects,type="response"),3)
```

- We add the column `pred` to our dataframe `BC`.
- `pred` is the final model's estimate of the probability of `malignant`.

```
head(BC[,c("Class","pred")])
```

```
##       Class  pred
## 1    benign 0.011
## 2    benign 0.945
## 3    benign 0.017
## 4    benign 0.929
## 5    benign 0.011
## 6 malignant 1.000
```

Not good for patients 2 and 4.

We may classify by `round(BC$pred)`:

- 0 to denote `benign` (probability `BC$pred` less than 0.5)
- 1 to denote `malignant` (probability `BC$pred` more than 0.5)

```
tally(~ Class + round(pred), data = BC)
```

```
##            round(pred)
## Class         0   1
##   benign    433  11
##   malignant  11 228
```

22 patients are misclassified.

```
sort(BC$pred[BC$Class=="malignant"])[1:5]
```

```
## [1] 0.035 0.037 0.089 0.190 0.205
```

There is a malignant woman with a predicted probability of malignancy, which is only 3.5%.

If we assign all women with predicted probability of malignancy above 5% to further investigation, then we only miss two malignant.

```
tally(~ Class + I(pred>.05), data = BC)
```

```
##            I(pred > 0.05)
## Class        TRUE FALSE
##   benign       50   394
##   malignant   237     2
```

The expense is that the number of false positive increases from 11 to 50.

```
tally(~ Class + I(pred>.1), data = BC)
```

```
##            I(pred > 0.1)
## Class        TRUE FALSE
##   benign       27   417
##   malignant   236     3
```

- If we instead set the alarm to 10%, then the number of false positives decreases from 50 to 27.
- But at the expense of 3 false negative.