Logistic regression - introduction

Ian Handel

2019-06-11

Introduction

These notes are a basic introduction to binary logistic regression used to analyse data with binary outcomes. In these notes we'll aim to cover the following learning outcomes:

- **Refresher** on logs, odds, probability and linear regression
- Understand why linear regression not sensible for **binary data**
- Explain how **logit** and binomial model let us **extend linear regression**
- Be able to run a **simple logistic regression in R**
- Be able to explain basic R glm **output**
- Be able to explain **estimates** with categorical and continuous variables
- Explain **significance test results** on variables
- Introduce some basic ideas for **selecting variables and models**
- **Things to watch out for!**
- **Know where to go next!**

Prerequisites

We'll cover a couple of background topics but to follow these notes you'll need to be able to load packages in R, run simple code in R and have a basic understanding of linear regression and statistical hypothesis tests.

Running the example code

To run the R code in these notes you'll need to start an rstudio project, load in the example data-set and have a few packages downloaded and loaded into your R session.

To download the packages (if you don't have them already) use...

```
install.packages("tidyverse")
install.packages("boot")
install.packages("broom")
install.packages("skimr")
install.packages("sjPlot")
```

Start a new project in rstudio and in an rscript use the floowing code to load the libraries you need and to import/load the data...

Loading the packages...

```
library(tidyverse)
library(boot)
library(broom)
library(skimr)
library(sjPlot)
```

Loading the data (from the csv file on Basecamp)...

```
dat <- read_csv("logreg_data_01_20190530.csv")
```

This dataset describes 500 animals giving their weight in kg, age in days, supplement levels (mg), sex, region where they liver (A, B, C or D) and whether they were treated with anthelmintics or not. It's a dataset made up for this course by the way!

Revision / background topics

Logarithms ('logs')

Skip this if you are happy with logs (including base 'e')

'Logs' are a mathematical function that changes a number. They take the form $log_b(x) = y$. What this means is b to the power y will give us x. It's easier to understand with examples. Let's start with base 10....

$$log_{10}(10) = 1$$

$$log_{10}(1000) = 3$$

$$log_{10}(0.01) = -2$$

So the logs of all the numbers in brackets are the number you'd need to raise 10 to to get them.

We can have other bases e.g. e., e is a special mathematical constant that feautres a lot behind the scene in statistics it's roughly 2.718...

$$log_e(2.718) \simeq 1$$

'Inverse logs' let us turn logs back into the original number. We simply raise the 'base' of our logs to the number we what to invert and we end up with the original number. So $log_{10}(1000) = 3$ and $10^3 = 1000$...

One feature of logs is that adding the logs of two numbers is equivalent to multiplying the numbers...

$$100 \times 1000 = 100000$$

$$log_{10}(100) + log_{10}(1000) = log_{10}(100000)$$

Because $log_{10}(100) = 2$, $log_{10}(1000) = 3$ and $log_{10}(100000) = 5$

If this all seems a bit too much don't worry. Just remember that adding logs is like multiplying numbers and you'll be fine!

Odds and probability

Probabilities have values from 0 ('never happens') to 1 ('always happens')

'events of interest' ÷ 'all events'

What is the probability that a fair coin lands on heads?

$$1/2 = 0.5$$

What is the probability that a 6 sided die lands on 4?

$$1/6 \simeq 0.166$$

Odds have values from 0 ('never happen') to infinity ('always happens')

'events of interest' ÷ 'other events'

What are the odds that a fair coin lands on heads?

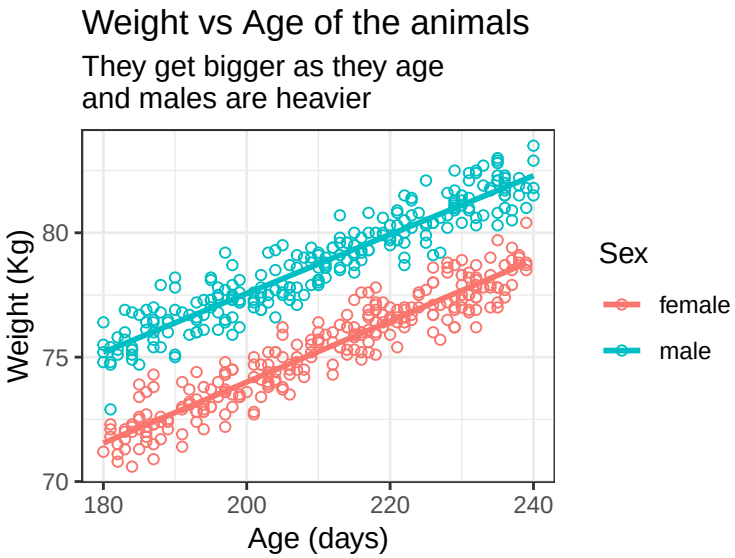
$$1/1 = 1$$

What are the odds that a 6 sided die lands on 4?

$$1/5 = 0.2$$

Linear regression

Remember that linear regression is a statistical method that lets us understand and predict numerical outcomes using one or more predictor variables. The predictor variables may be numerical or categorical. Linear regression also assumes there's a linear i.e. straight line relationship between the predictor numerical variable sand the outcome. Using the data set we have loaded...



In R we can use the `lm()` function to fit linear models. Normally we would store the results of using the function in an R object and look at it using 'summary(). We can also get a tidier output using `get_model_data()` from the `sjPlot` package. Here we make a linear model predicting the animal's weight from their age (a numerical variable) and their sex (a categorical variable).

```
mod1 <- lm(weight ~ age + sex, data = dat)
```

Using `summary()` from base-R...

```
summary(mod1)
```

```
Call:
lm(formula = weight ~ age + sex, data = dat)

Residuals:
    Min      1Q  Median      3Q     Max
-2.3506 -0.4732 -0.0509  0.4572  2.0223

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 49.93305    0.40486  123.33   <2e-16 ***
age          0.12045    0.00191   63.07   <2e-16 ***
sexmale      3.51683    0.06541   53.76   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.7312 on 497 degrees of freedom
Multiple R-squared:  0.932, Adjusted R-squared:  0.9317
F-statistic:  3404 on 2 and 497 DF,  p-value: < 2.2e-16
```

Using `get_model_data()` from `sjPlot` package and selecting the output columns we want...

```
get_model_data(mod1) %>%
  select(term:p.stars) %>%
  print()
```

```
# A tibble: 2 x 8
  term    estimate std.error statistic   p.value conf.low conf.high p.stars
  <fct>      <dbl>     <dbl>     <dbl>     <dbl>    <dbl>     <dbl> <chr>
1 age        0.120   0.00191      63.1 2.50e-239    0.117     0.124 ***
2 sexmale    3.52    0.0654       53.8 2.87e-209    3.39      3.65  ***
```

The analysis suggestes that both **age** and **sex** are significant predcitors of weight. Weight increasing by about 0.12 Kg per day of age and that males are about 3.5 Kg heavier than females.

Analysing binary data

Binary data common in epidemiology e.g.

- alive/dead
- healthy/diseased

ID	treatment	age	region	supp	sex	weight	status
A0049	control	221	D	7.891	male	80.9	diseased
A0485	control	220	A	2.651	male	79.5	diseased
A0321	treated	238	B	1.671	male	82.2	diseased
A0153	treated	183	C	6.346	female	71.7	healthy
A0074	treated	187	A	9.655	male	77.0	diseased
A0228	control	206	C	8.046	female	74.7	diseased

Univariable analysis

Status vs treatment

treatment	diseased	healthy
control	154	44
treated	209	93

```
with(dat,
     {{fisher.test(status, treatment)}})
```

```

    Fisher's Exact Test for Count Data

data:  status and treatment
p-value = 0.04032
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 1.010694 2.419528
sample estimates:
odds ratio
  1.556051
```

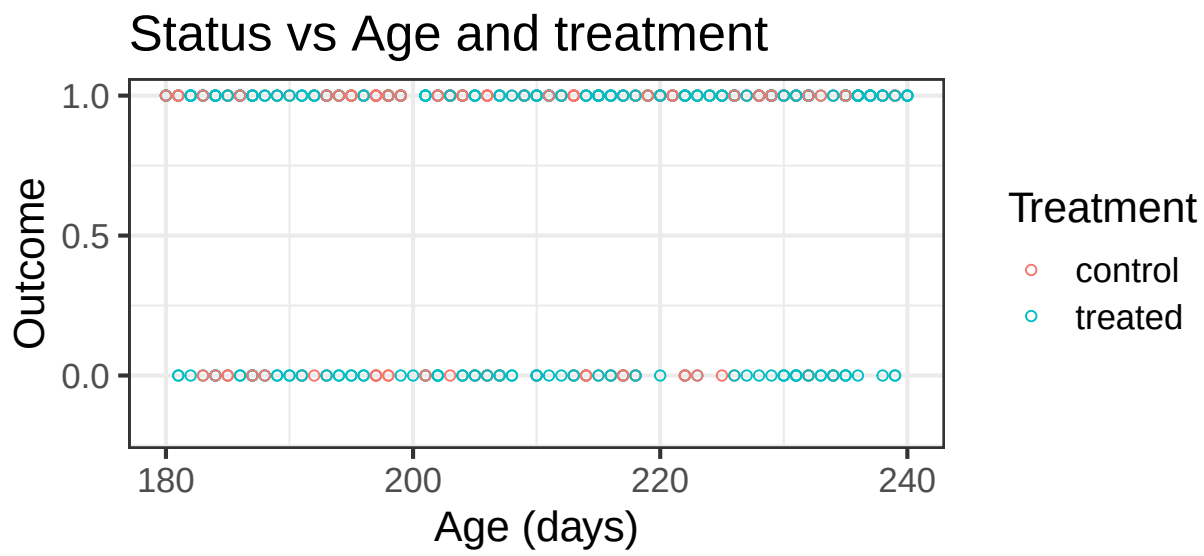
Multivariable analysis

How about recoding the outcome as 0/1?

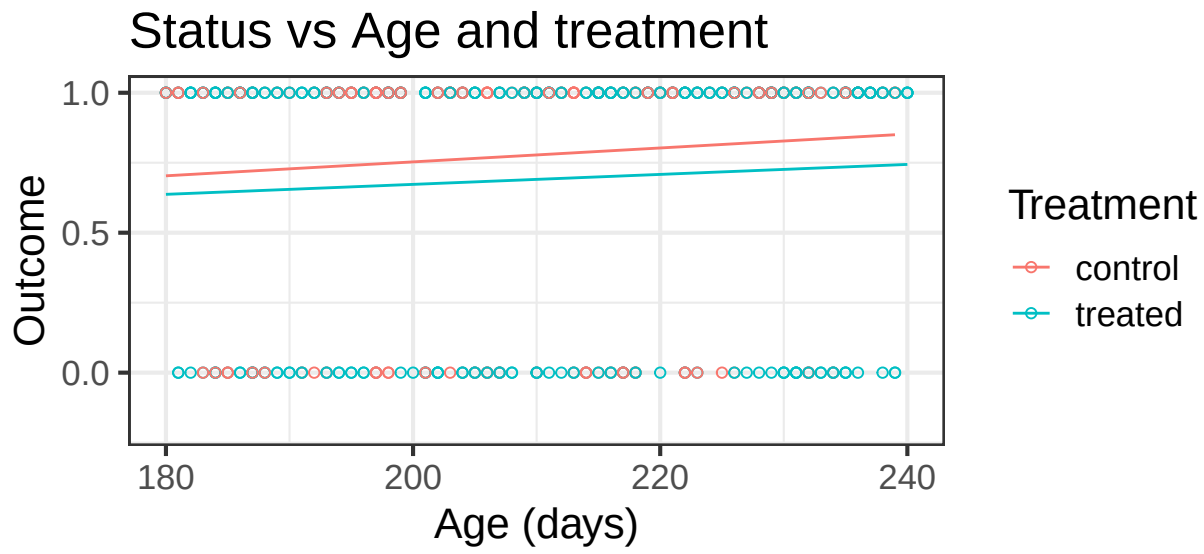
ID	treatment	age	region	supp	sex	weight	status	status01
A0049	control	221	D	7.891	male	80.9	diseased	1
A0485	control	220	A	2.651	male	79.5	diseased	1
A0321	treated	238	B	1.671	male	82.2	diseased	1
A0153	treated	183	C	6.346	female	71.7	healthy	0
A0074	treated	187	A	9.655	male	77.0	diseased	1
A0228	control	206	C	8.046	female	74.7	diseased	1

Then use linear regression...

Linear regression 1



Linear regression 2



Problems

-predicts (impossible) intermediate values

-can predict <0 and >1

So how do we fix this?

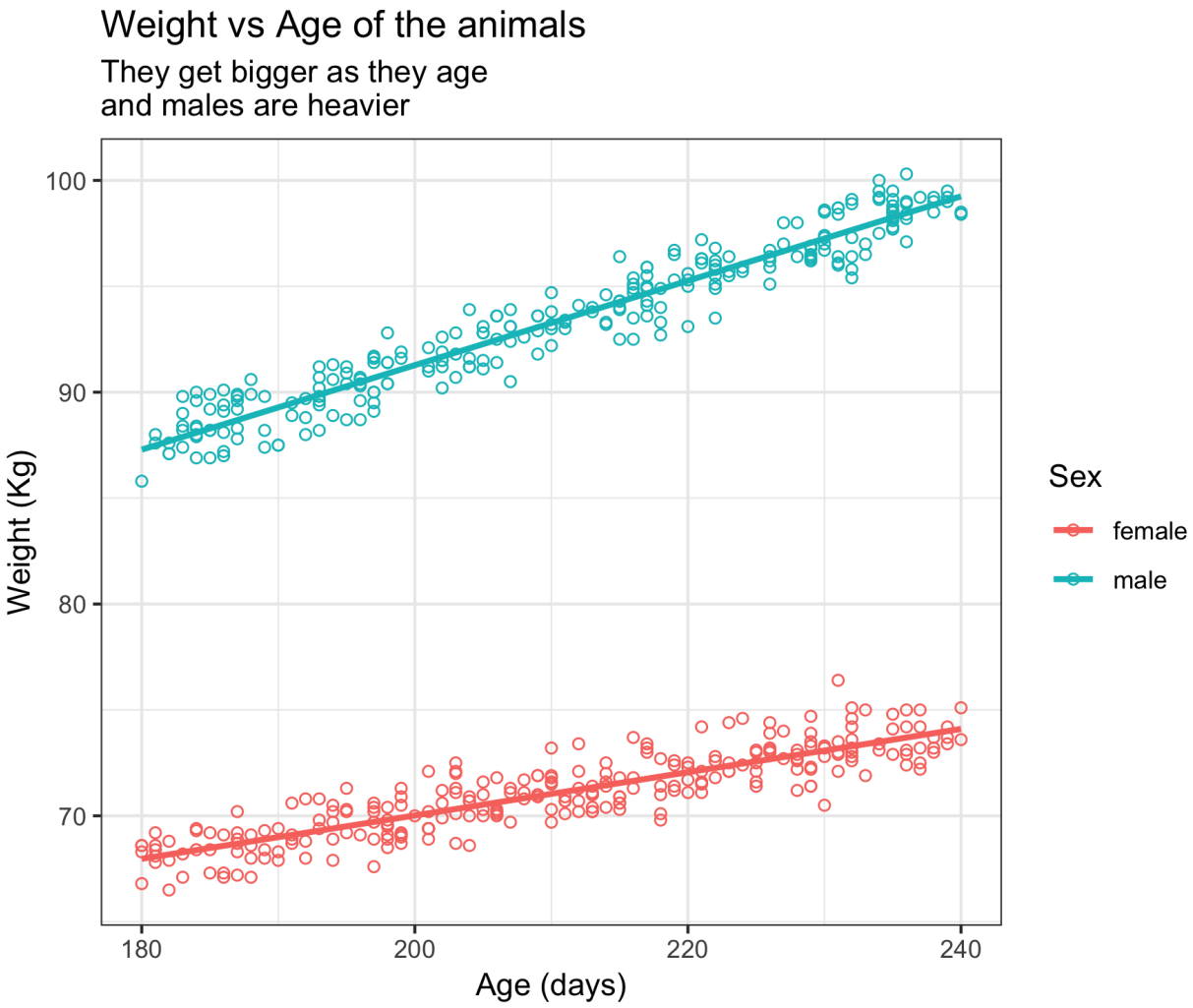
Linear regression does this...

$weight \sim \beta_0 + \beta_1 age + \beta_2 sex + \epsilon$

or in english...

The outcome, *weight*, is related to the predictors

by one or more straight lines.



For binary data we want

Our outcome to be 0 or 1

So rather than modelling the outcome.

We model the **probability** of something e.g. being diseased...

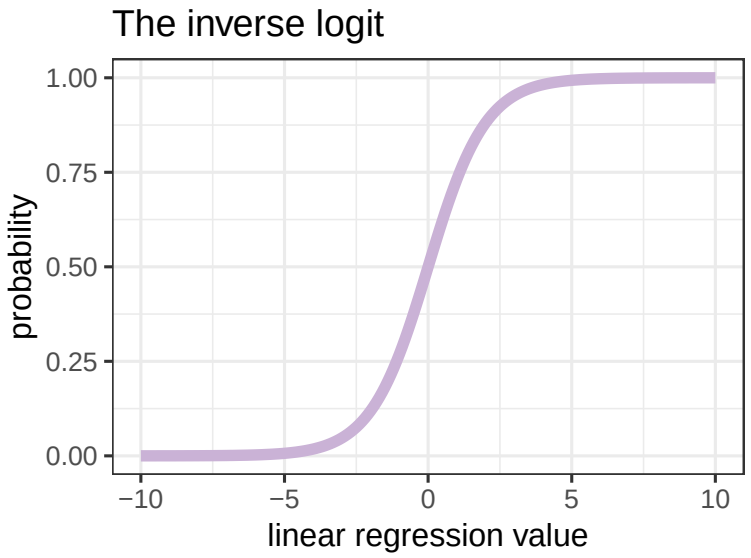
The logistic bit...

Linear regression models model numbers, any numbers!

Probabilities go from...

0 to 1

So we need to turn any number into 0 - 1



In fact the regression value is the log of the odds of the outcome.

#The logistic bit 2

So we have an outcome, e.g. being diseased vs healthy, that is coded 0 or 1

And our model is

$$log_e(\frac{prob}{1-prob}) \sim \beta_0 + \beta_1 age + \beta_2 treatment)$$

or in english

The log of the odds of an animal being diseased are modelled by a linear combination of the predictor variables

class: inverse, middle, center

Worked example in R

R code for logistic regression

```
head(dat)
```

ID	treatment	age	region	supp	sex	weight	status	status01
A0001	treated	219	D	6.731	female	76.2	diseased	1
A0002	treated	218	C	7.950	female	76.0	diseased	1
A0003	control	214	A	5.617	female	75.3	healthy	0
A0004	control	194	A	8.490	female	72.1	diseased	1
A0005	treated	185	D	7.127	female	72.6	healthy	0
A0006	treated	235	A	4.882	female	77.8	healthy	0

A linear model of weight

```
mod_weight <- lm(weight ~ age + sex, data = dat)
```

A logistic regression model of disease status

```
mod_disease <- glm(status01 ~ treatment + age, family = binomial, data = dat)
```

The output

```
print(summary(mod_disease), digits = 3)
```

```

Call:
glm(formula = status01 ~ treatment + age, family = binomial,
    data = dat)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-1.847  -1.428   0.746   0.826   0.973

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)      -0.88412    1.23913   -0.71    0.476
treatmenttreated -0.45345    0.21225   -2.14    0.033 *
age               0.01021    0.00589    1.74    0.083 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 587.20  on 499  degrees of freedom
Residual deviance: 579.67  on 497  degrees of freedom
AIC: 585.7

Number of Fisher Scoring iterations: 4
```

The output

```
print(summary(mod_disease), digits = 3)
```

```

Call:
glm(formula = status01 ~ treatment + age, family = binomial,
    data = dat)

Deviance Residuals:
```

```
   Min      1Q  Median      3Q     Max
-1.847  -1.428   0.746   0.826   0.973

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)      -0.88412    1.23913   -0.71    0.476
treatmenttreated -0.45345    0.21225   -2.14    0.033 *
age               0.01021    0.00589    1.74    0.083 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 587.20  on 499  degrees of freedom
Residual deviance: 579.67  on 497  degrees of freedom
AIC: 585.7

Number of Fisher Scoring iterations: 4
```

The output

Lets get 'tidy output...

```
tidy(mod_disease) #tidy from the broom package
```

odds ratios

The estimates = log(odds ratios)

i.e.

$$\frac{odds\ of\ outcome\ if\ have\ factor}{odds\ of\ outcome\ if\ dont\ have\ factor}$$

So we get odds ratios by 'inverse logging them'.

We can remove the intercept.

```
tidy(mod_disease) %>%
  filter(term != "(Intercept)") %>%
  mutate(OR = exp(estimate))
```

A results table

```
tidy(mod_disease) %>%
  mutate(OR = exp(estimate)) %>%
    bind_cols(exp(confint_tidy(mod_disease)) %>%
    as_tibble()
```

```
) %>%
filter(term != "(Intercept)") %>%
select(term, OR, `conf.low`, `conf.high`, p.value)
```

term	OR	conf.low	conf.high	p.value
treatmenttreated	0.635434	0.4165578	0.9586278	0.0326443
age	1.010266	0.9987142	1.0220622	0.0827082

But what does it mean?

Interpreting the odds ratios

term	OR	2.5 %	97.5 %	p.value
treatmenttreated	0.635434	0.4165578	0.9586278	0.0326443
age	1.010266	0.9987142	1.0220622	0.0827082

Odds ratios multiply

Categorical predictors

How many times greater the odds of outcome are **if** the risk factor (etc) is present.

So for the treatment variable (which can be control or treatment) the odds of disease if treated are 0.635 **times greater** than if untreated (control).

Interpreting the odds ratios

term	OR	2.5 %	97.5 %	p.value
treatmenttreated	0.635434	0.4165578	0.9586278	0.0326443
age	1.010266	0.9987142	1.0220622	0.0827082

Odds ratios multiply

Numerical predictors

How many times greater the odds of outcome are for **each unit change** in the variable

So for the age variable the odds of disease are 1.01 **times greater** for each day older.

So for 3 days it's 1.01 x 1.01 x 1.01 $\simeq$ 1.03.

Things to watch out for

Factor levels

How does R know if you are predicting 'healthy' or 'diseased'?

Perfect predictors

E.g. all the males are diseased and all the females are healthy

Linear on logit

Disease risk might go up and then down

Model selection - a blank page

More help

Veterinary Epi Reasearch - Ian Dahoo