Logistic regression - an introdution

Ian Handel
June 2019

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
- Things to watch out for!
- **Know where to go next!**

But first - some R

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

dat <- read_csv("logreg_data_01_20190530.csv")

skim(dat)
```

Logarithms ('logs')

Can we skip this bit?

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

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

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

We can have other bases e.g. e

$$log_e(2.718) \simeq 1$$

And reversing this...

$$10^3 = 1000$$

$$e^2 \simeq 7.389$$

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 is the odds that a fair coin lands on heads?

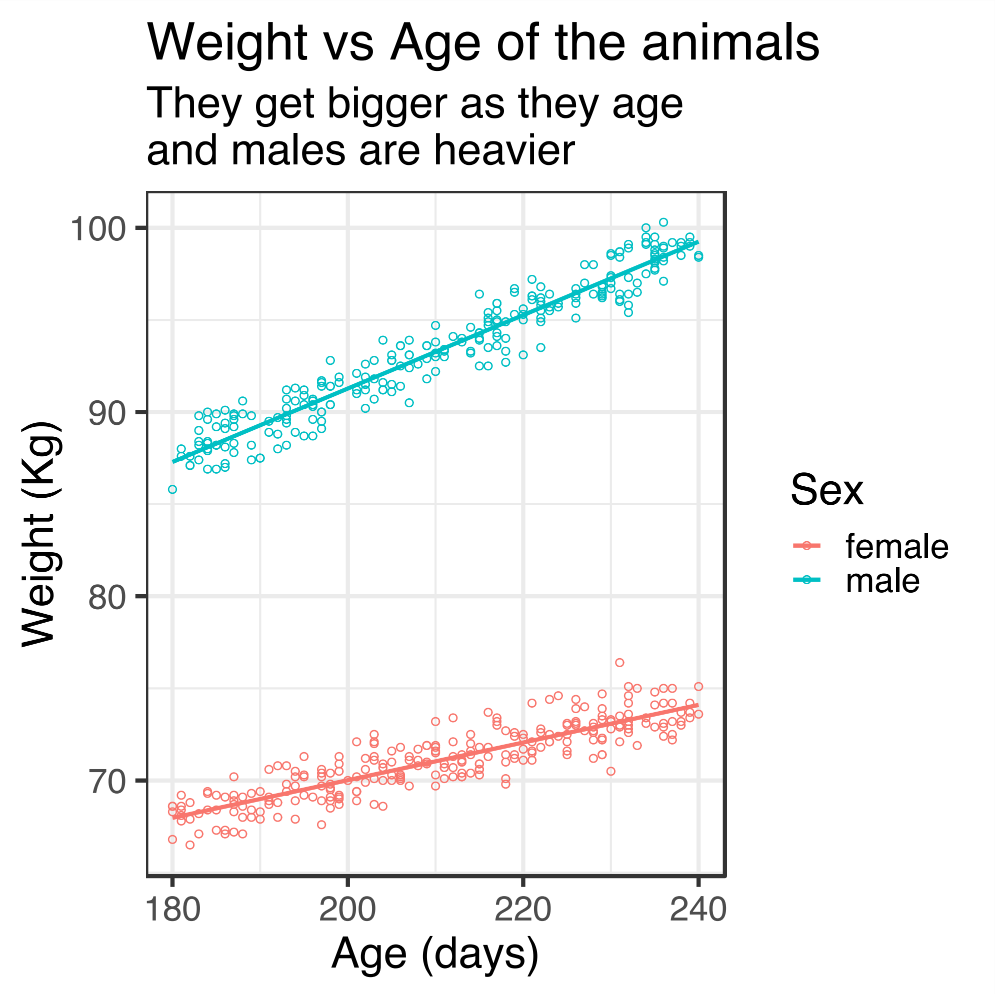
$$1/1 = 1$$

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

$$1/5 = 0.2$$

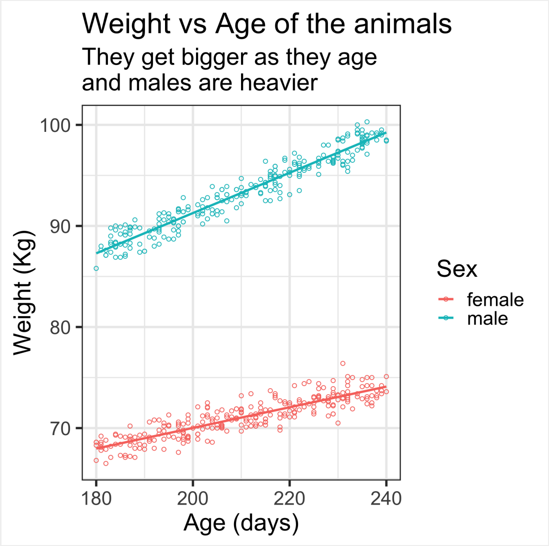
Linear regression

- numerical outcome
- numerical / categorical predictors
- linear relationship



Linear regression in R

```
mod ← lm(weight ~ age + sex, data = dat
```




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

Residuals:
   Min      1Q  Median      3Q     Max
 -3.54   -0.88   -0.02    0.89    3.07

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  39.1901     0.7026      56   <2e-16 ***
age           0.1515     0.0033      46   <2e-16 ***
sexmale      22.2796     0.1136     196   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.3 on 497 degrees of freedom
Multiple R-squared:  0.99,    Adjusted R-squared:  0.99
F-statistic: 2e+04 on 2 and 497 DF,  p-value: <2e-16
```

Analysing binary data

Binary data common in epidemiology e.g.

- alive/dead
- healthy/diseased

Example data

ID	treatment	age	region	sex	weight	status
A0458	control	209	C	female	70.9	healthy
A0468	treated	190	C	female	68.3	healthy
A0143	control	239	B	female	73.7	diseased
A0413	control	235	D	male	97.8	healthy
A0319	control	197	B	male	89.1	healthy
A0257	control	194	B	female	69.7	healthy

Univariable analysis

Status vs treatment

treatment	diseased	healthy
control	43	168
treated	34	255

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

```

    Fisher's Exact Test for Count Data

data:  status and treatment
p-value = 0.01175
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 1.142387 3.238178
sample estimates:
odds ratio
  1.917107
```

Multivariable analysis

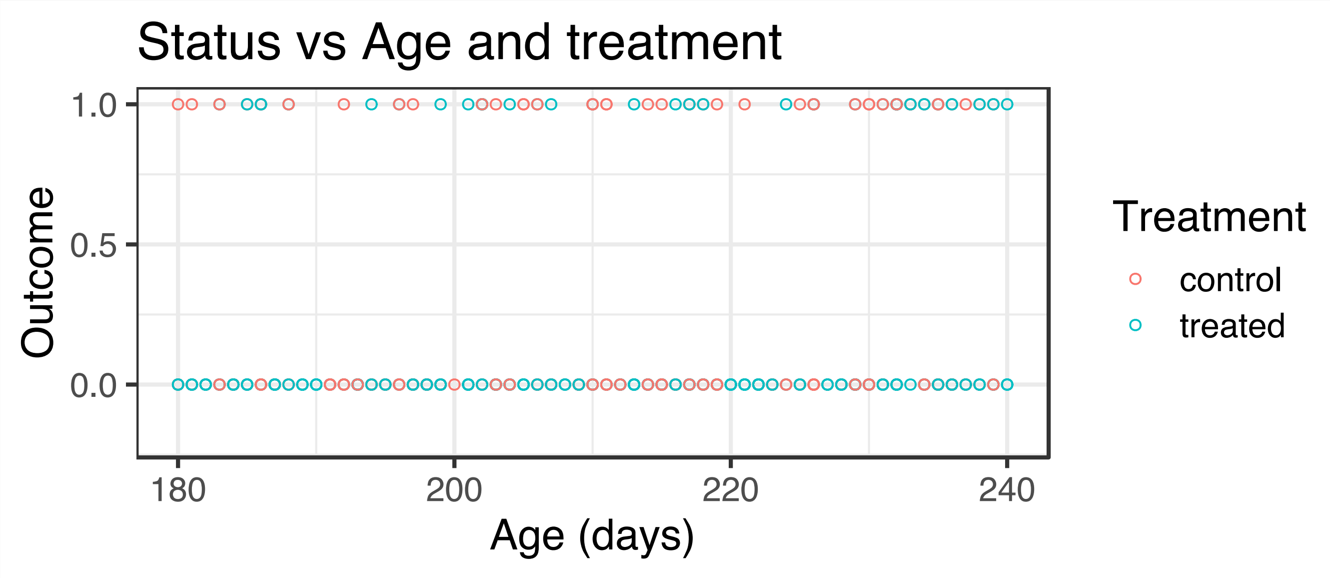
How about recoding the outcome as 0/1?

Example data

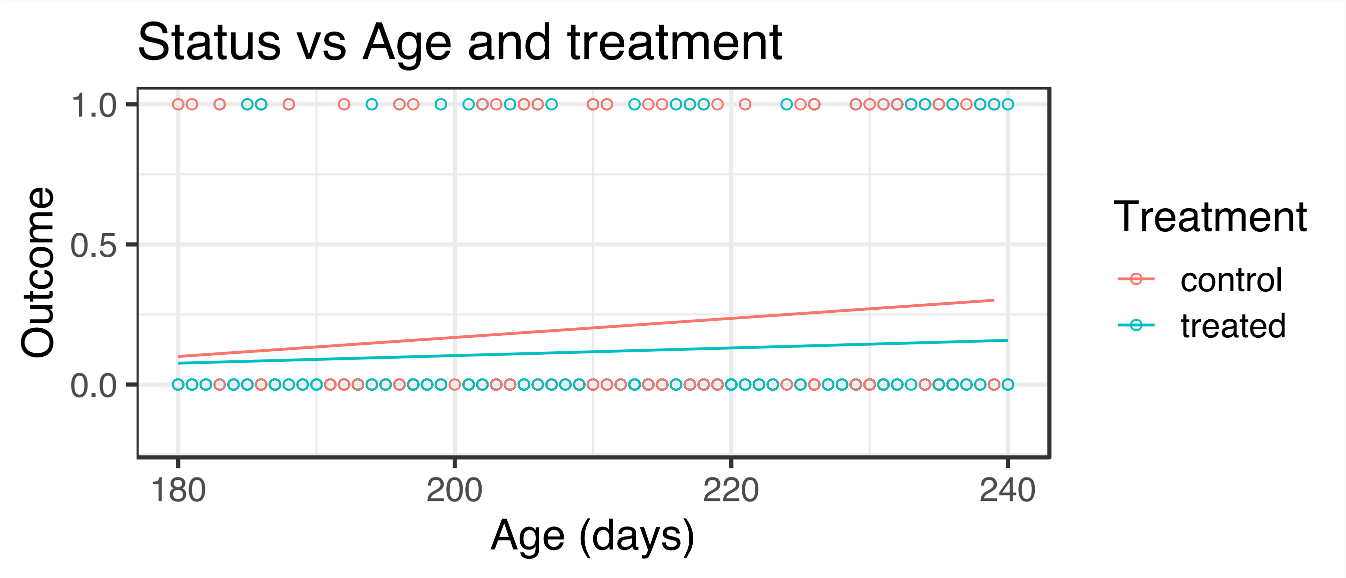
ID	treatment	age	region	sex	weight	status	status01
A0458	control	209	C	female	70.9	healthy	0
A0468	treated	190	C	female	68.3	healthy	0
A0143	control	239	B	female	73.7	diseased	1
A0413	control	235	D	male	97.8	healthy	0
A0319	control	197	B	male	89.1	healthy	0
A0257	control	194	B	female	69.7	healthy	0

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

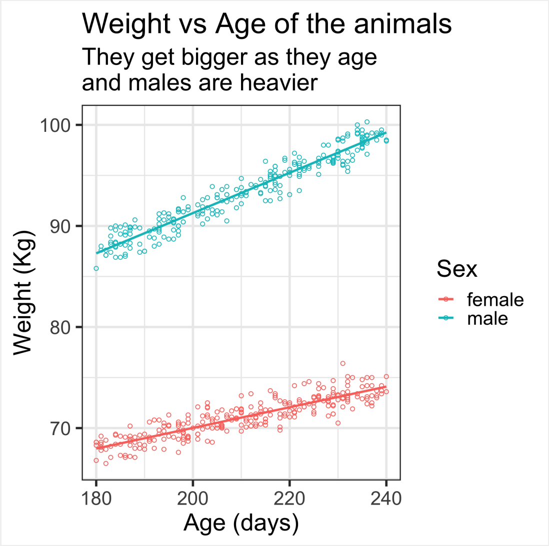
by one or more straight lines.

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...



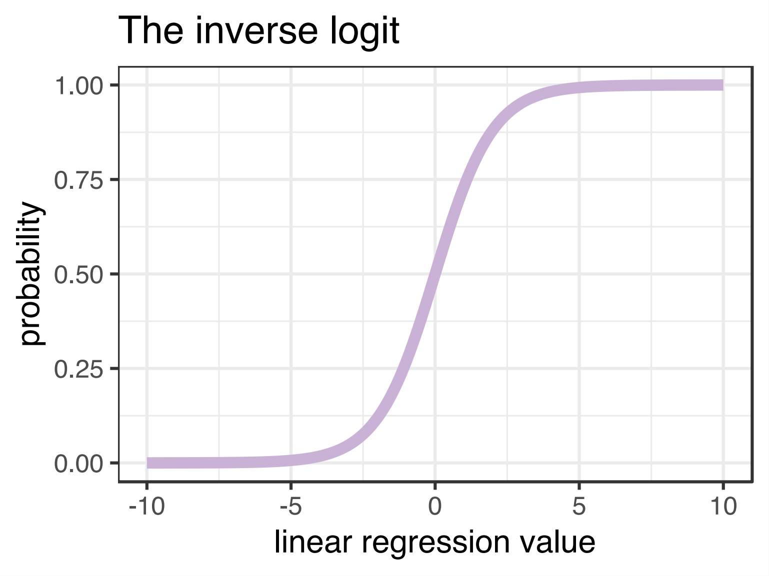
The logistic bit...

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

Worked example in R

R code for logistic regression

```
head(dat)
```

ID	treatment	age	region	sex	weight	status	status01
A0001	control	219	A	female	71.4	diseased	1
A0002	control	218	A	female	70.1	healthy	0
A0003	treated	214	D	female	71.4	healthy	0
A0004	treated	194	D	female	68.9	healthy	0
A0005	control	185	D	female	67.3	healthy	0
A0006	treated	235	D	male	98.6	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

```
summary(mod_disease)
```

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

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.8300  -0.6054  -0.5320  -0.4210   2.2841

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)      -5.074379   1.613819  -3.144  0.00166 **
treatmenttreated -0.662589   0.251643  -2.633  0.00846 **
age               0.017514   0.007521   2.329  0.01987 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 429.59  on 499  degrees of freedom
Residual deviance: 417.16  on 497  degrees of freedom
AIC: 423.16

Number of Fisher Scoring iterations: 4
```

The output

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

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

Deviance Residuals:
   Min       1Q   Median       3Q      Max
-0.830   -0.605   -0.532   -0.421    2.284

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)      -5.07438    1.61382   -3.14   0.0017 **
treatmenttreated -0.66259    0.25164   -2.63   0.0085 **
age               0.01751    0.00752    2.33   0.0199 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 429.59  on 499  degrees of freedom
Residual deviance: 417.16  on 497  degrees of freedom
AIC: 423.2

Number of Fisher Scoring iterations: 4
```

The output

Lets get 'tidy output...

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

```
# A tibble: 3 x 5
  term             estimate std.error statistic p.value
  <chr>               <dbl>     <dbl>     <dbl>   <dbl>
1 (Intercept)       -5.07     1.61        -3.14 0.00166
2 treatmenttreated  -0.663    0.252       -2.63 0.00846
3 age                0.0175   0.00752      2.33 0.0199
```

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 tibble: 2 x 6
  term             estimate std.error statistic p.value    OR
  <chr>               <dbl>     <dbl>     <dbl>   <dbl> <dbl>
1 treatmenttreated  -0.663    0.252       -2.63 0.00846 0.516
2 age                0.0175   0.00752      2.33 0.0199  1.02
```

A results table

```
tidy(mod_disease) %>%
  mutate(OR = exp(estimate)) %>%
    bind_cols(exp(confint(mod_disease)) %>%
    as_tibble()
  ) %>%
  filter(term != "(Intercept)") %>%
  select(term, OR, `2.5 %`, `97.5 %`, p.value)
```

term	**OR**	**2.5 %**	**97.5 %**	**p.value**
treatmenttreated	0.516	0.313	0.842	0.008
age	1.018	1.003	1.033	0.020

But what does it mean?

Interpreting the odds ratios

term	OR	2.5 %	97.5 %	p.value
treatmenttreated	0.516	0.313	0.842	0.008
age	1.018	1.003	1.033	0.020

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 0.516 **times greater** than if untreated (control).

Interpreting the odds ratios

term	OR	2.5 %	97.5 %	p.value
treatmenttreated	0.516	0.313	0.842	0.008
age	1.018	1.003	1.033	0.020

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 1.018 **times greater** for each day older.

So for 3 days it's 1.018 x 1.018 x 1.018 $\simeq$ 1.055.

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

More help

Dohoo book