Model fitting

Thomas Lumley
Ken Rice

Model Fitting

Seattle, June 2009

Regression commands

Two of the most important `R` commands;

- `lm()`: fits **L**inear **M**odels
- `glm()`: fits **G**eneralized **L**inear **M**odels

(If you've used SAS, its `glm` is **not** the same as `R`'s)

'Linear Regression' and 'Logistic Regression' are special cases.

The help files are huge (and generic) – how are `lm()`, `glm()` used in genetics?

Use of `lm()` in genetics

For a continuous outcome,

`lm(outcome ~ genetic.predictor, [...] )`

estimates the association between `outcome` and `predictor`

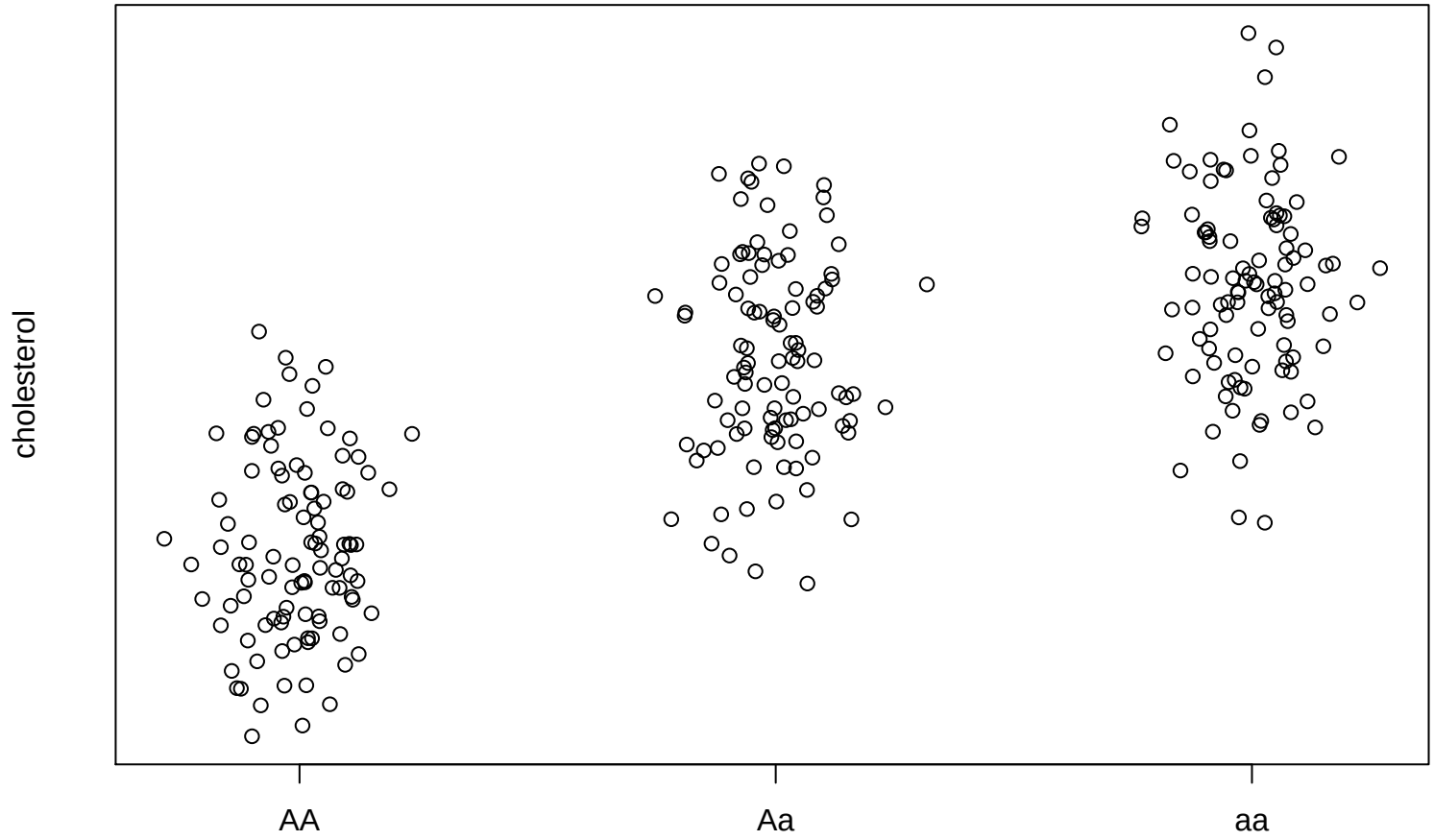
The **optional** arguments [...] might be

- `data = my.data` – your dataset
- `subset = race=="CEPH"` – use partial data
- `weights =` – for advanced analyses

Model Description	`predictor`	Common name
Number of minor alleles	`(g=='Aa') + 2*(g=='aa')` or `as.numeric(g)`	Additive
Presence of minor allele	`(g=='Aa') | (g=='aa')`	Dominant
Homozygous for minor allele	`g=='aa'`	Recessive
Distinct effects for hetero/homozygous	`factor(g)`	2 parameter, or "2 df"

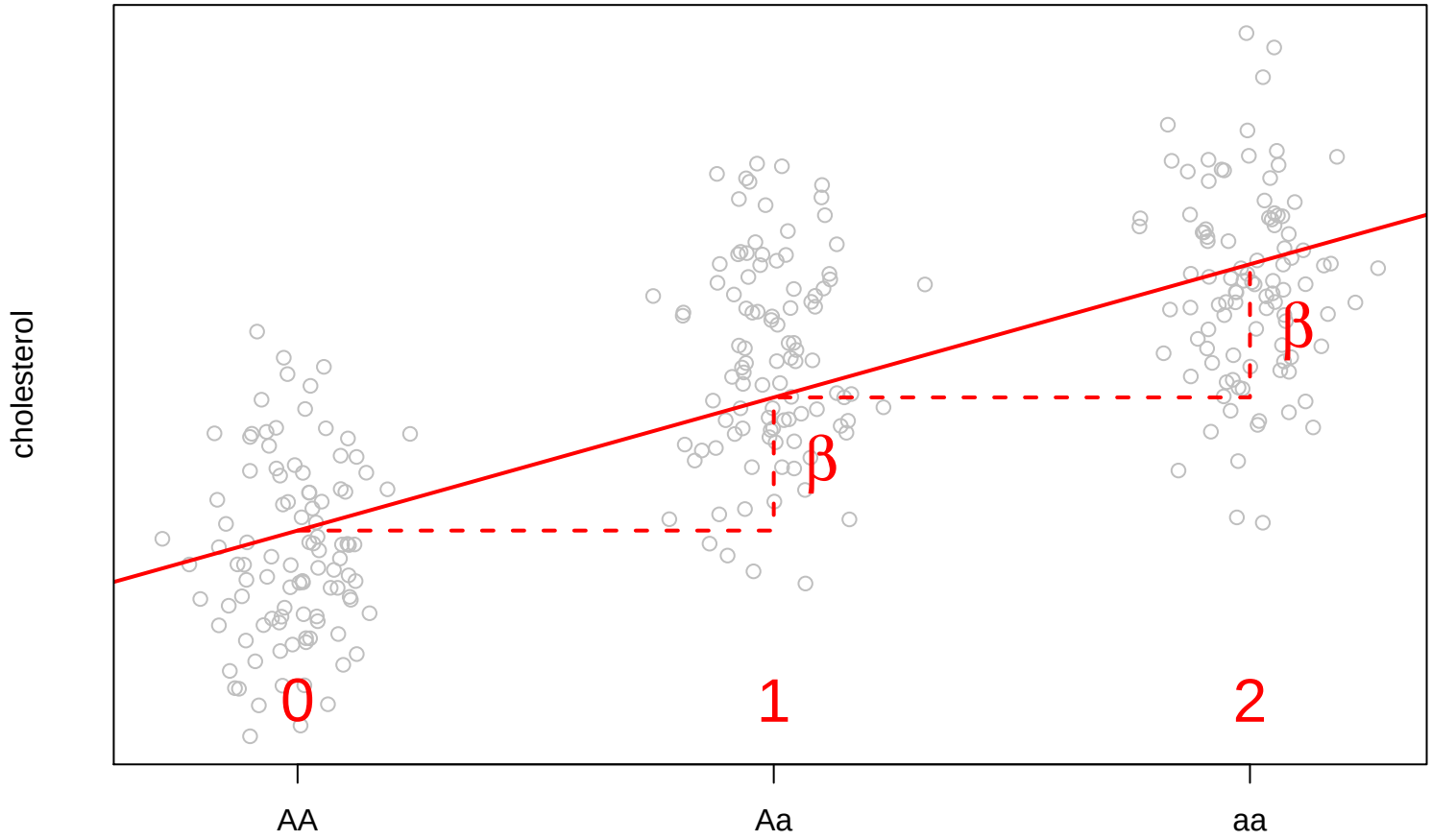
Use of `lm()` in genetics

Some data; cholesterol levels plotted by genotype (single SNP)



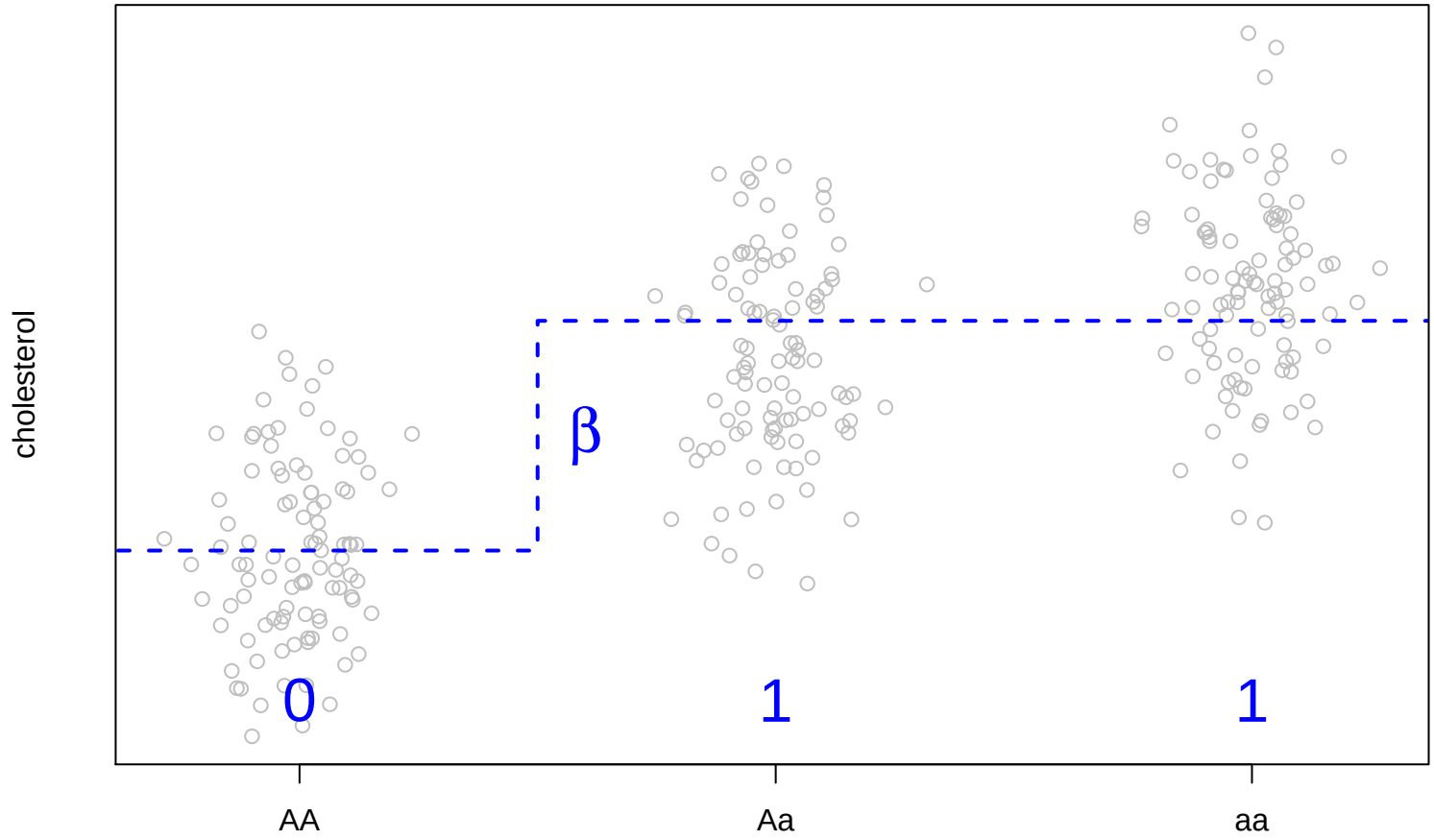
Use of `lm()` in genetics

Additive model (the most commonly used)



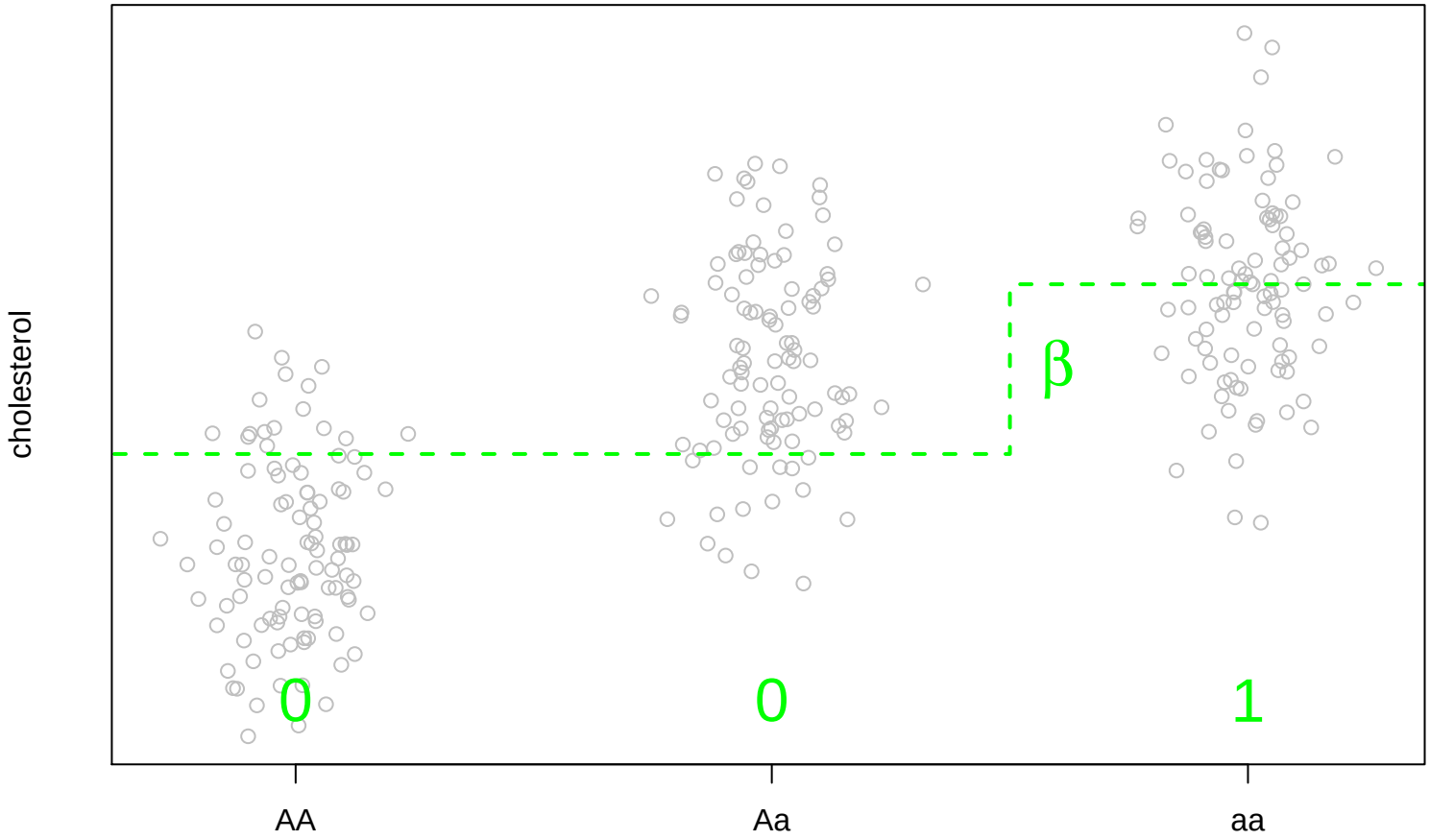
Use of `lm()` in genetics

Dominant model (best fit to this data)



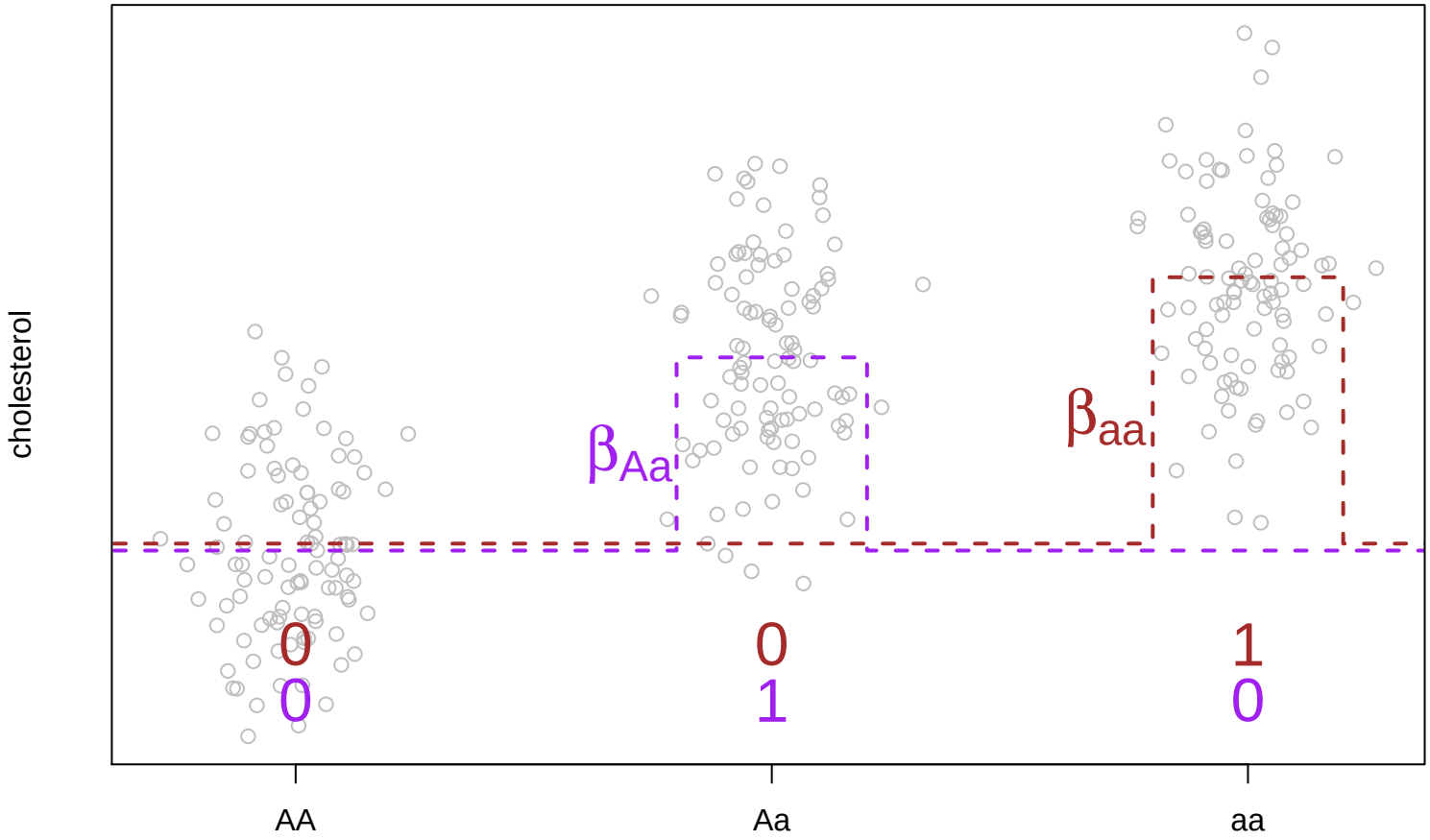
Use of `lm()` in genetics

Recessive model (least stable for rare `aa`)



Use of `lm()` in genetics

2 parameter model (robust but can be overkill)



`lm()`: Estimates, Intervals, p-values

`lm()` produces **point estimates** for your model;

```
> predictor <- (g=="Aa") + 2*(g=="aa") #the number of 'a' alleles
> my.lm <- lm( cholesterol ~ predictor )
> my.lm
Call:
lm(formula = cholesterol ~ predictor)
Coefficients:
(Intercept)      predictor
     0.2104        0.9507
```

– also available via `my.lm$coefficients`.

The `coefficients` in the output tell you the **additive increase** in `outcome` associated with a **one-unit** difference in the `genetic predictor`.

The coefficient for `predictor` is in units of `cholesterol`

`lm()`: Estimates, Intervals, p-values

You will also want **confidence intervals**;

```
> confint.default(my.lm)
                 2.5 %    97.5 %
(Intercept) 0.08391672 0.3368275
predictor   0.85279147 1.0486953
```

Remember to **round these numbers** to an appropriate number of significant figures! (2 or 3 is usually enough)

We are **seldom** interested in the `Intercept`

`lm()`: Estimates, Intervals, p-values

Two-sided **p-values** are also available;

```
> summary(my.lm)
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.21037    0.06426   3.274  0.00119 **
predictor    0.95074    0.04977  19.101  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

In this data, we have **strong evidence** of an **additive effect** of the minor allele on cholesterol

`summary(my.lm)` gives **many** other details – ignore for now

Confidence intervals are just `Estimate` $\pm$ 2$\times$`Std.Error`

Use of `glm()` in genetics

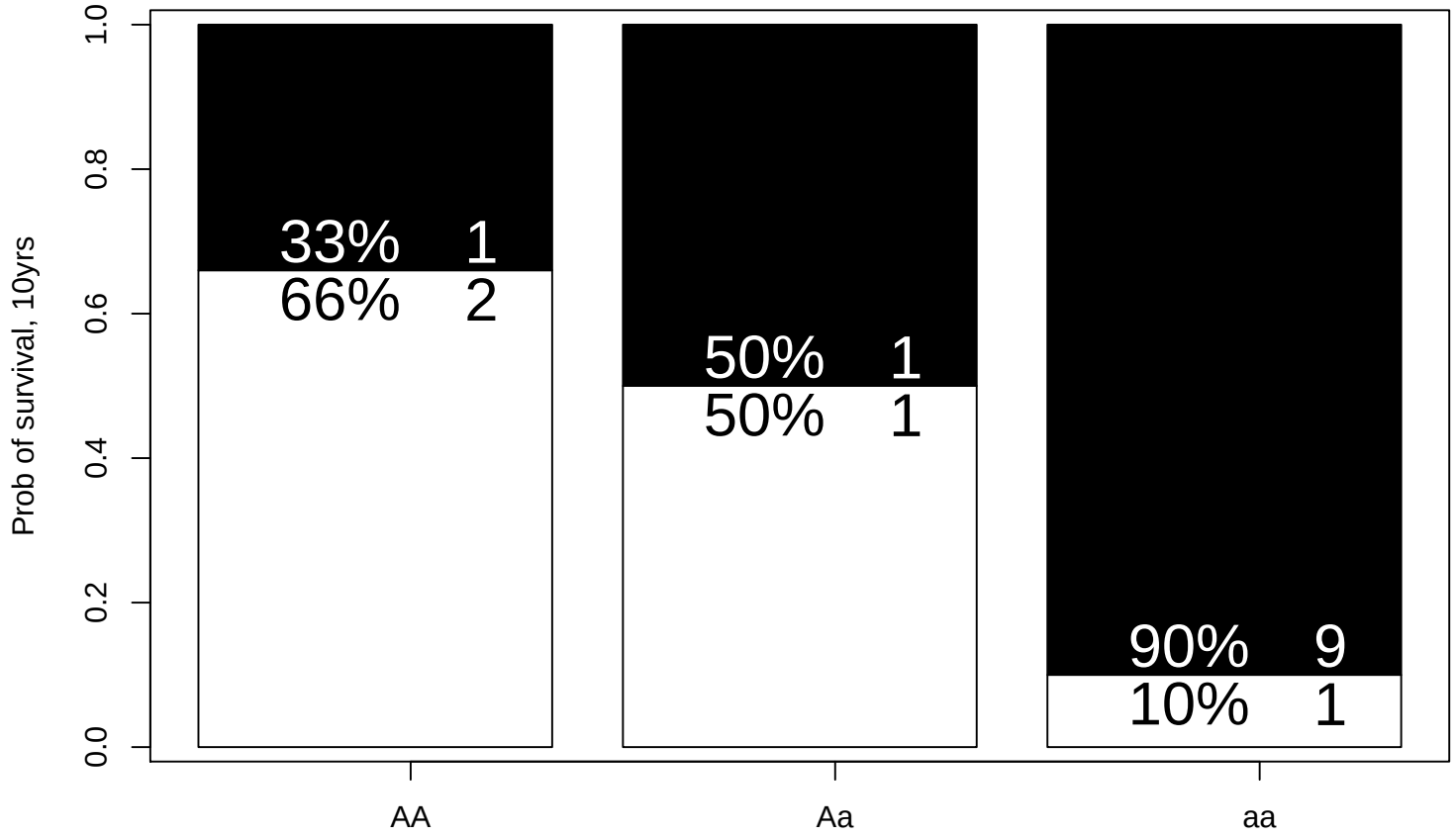
Logistic regression is the 'default' analysis for **binary outcomes**

Outcome	Type	Regression	Scale
Cholesterol Blood Pressure BMI	Continuous	Linear	Difference in Outcome
Death Stroke BMI>30	Binary	Logistic	Ratio of odds

What are **odds**? Really just **probability**...

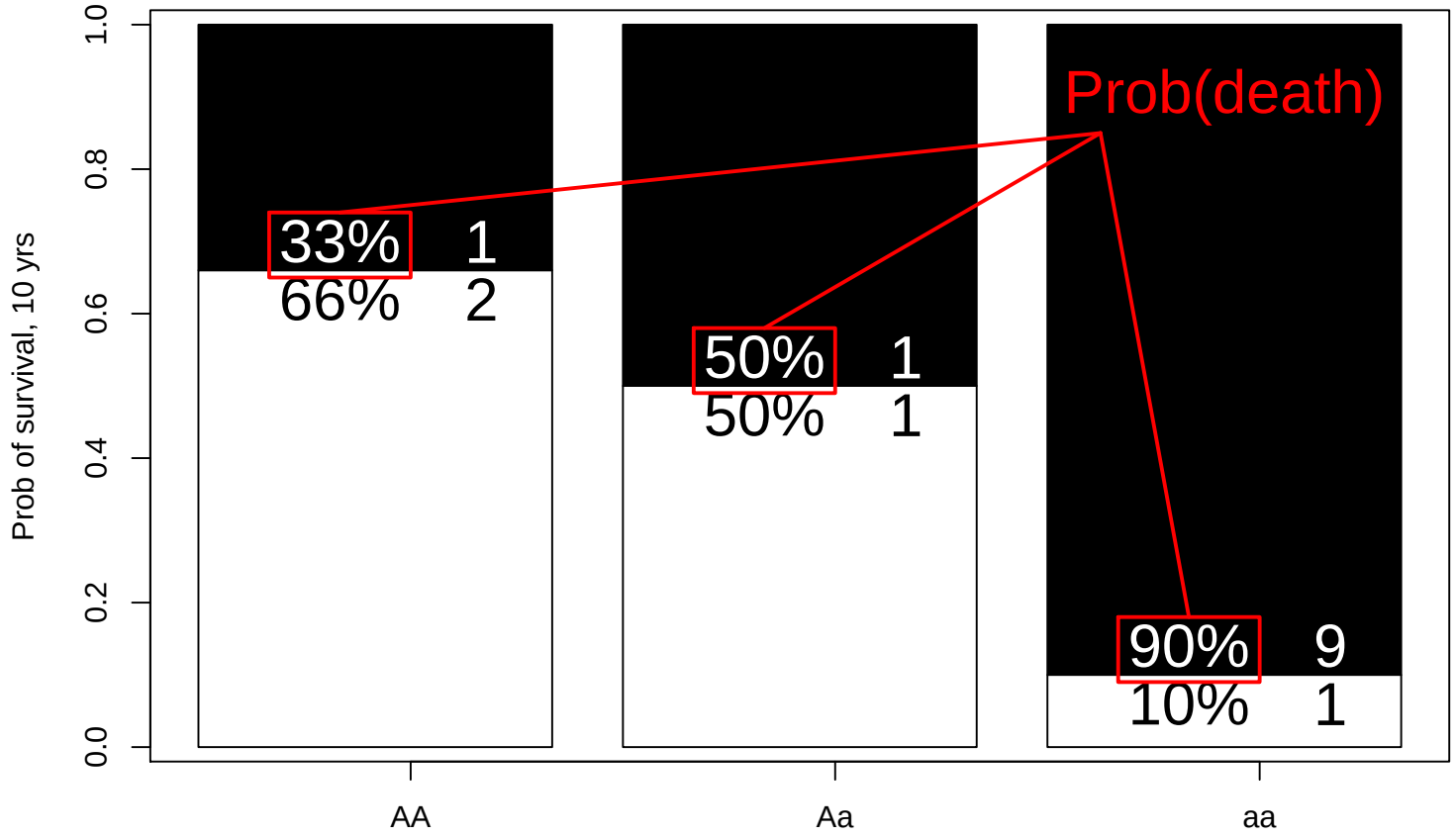
Use of `glm()` in genetics

Odds are a [gambling-friendly] measure of chance;



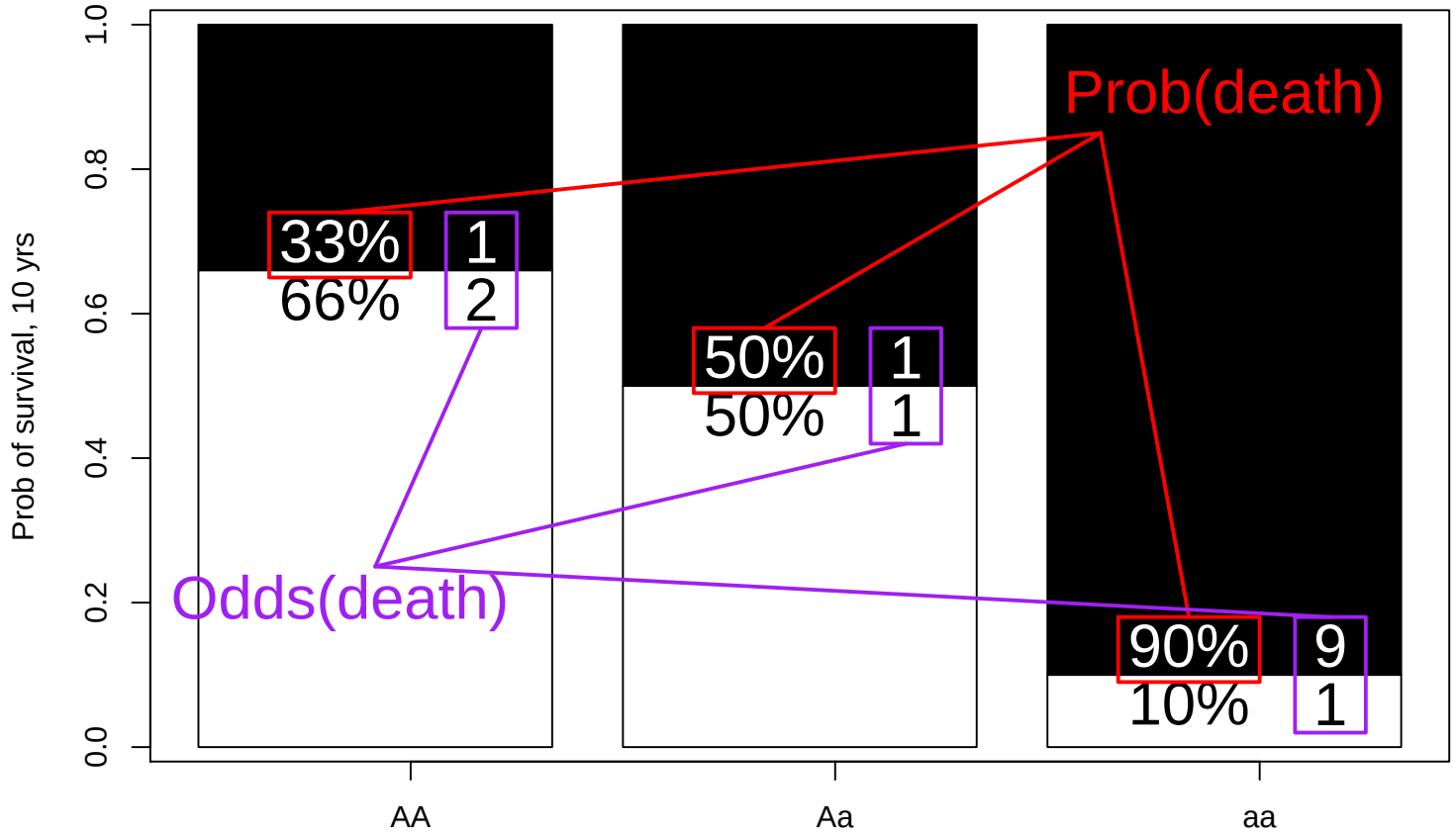
Use of `glm()` in genetics

Odds are a [gambling-friendly] measure of chance;



Use of `glm()` in genetics

Odds are a [gambling-friendly] measure of chance;



– so what are **odds ratios**?

Use of `glm()` in genetics

Using the data from the bar charts;

```
> pred2 <- factor(g)
> glm1 <- glm( dead10yrs ~ pred2, family=binomial)
> glm1
     pred2Aa      pred2aa
      0.6931       2.8904
```

These are **log odds ratio** estimates; to transform to OR, use $e^{0.6931} = 2, e^{2.8904} = 18$

They are given **relative to the baseline group** – 'AA' in this case

Don't forget the `family=binomial` argument!

Use of `glm()` in genetics

Confidence intervals, p-values as with `lm()`, **for the log odds ratios**;

```
> confint.default(glm1)
                2.5 %     97.5 %
pred2Aa     0.1201986  1.2660957
pred2aa     2.1148912  3.6658523
```

```
> summary(glm1)
            Estimate Std. Error z value Pr(>|z|)
pred2Aa       0.6931     0.2923   2.371  0.01773 *
pred2aa       2.8904     0.3957   7.305 2.77e-13 ***
```

Use `exp()` to get odds ratio estimates, intervals; p-values are **scale-independent**

The formula syntax

We fit `lm(y~ predictor)` and `glm(y~ pred2)`. To see how phenotype depends on *several* covariates, we specify e.g.

```
y ~ genotype.pred + age + sex
```

– formally, this gives *multivariate regression*; the `genotype.pred` coefficients reflect the genotype effects *adjusted for age and sex*

- Separate covariates with '+'. This is *not* addition!
- For now, make predictor variables first, then do regression; doing everything in one step is possible, but requires care when using e.g. addition (see above)
- For keen people; in the formula syntax, `*` indicates that interactions should be fitted, `I()` insulates mathematical operations, `-1` removes the intercept... see `?formula`