4. Model fitting

Thomas Lumley
Ken Rice

Universities of Washington and Auckland

Seattle, July 2013

Disclaimer

We can't teach regression in one session. But we will cover;

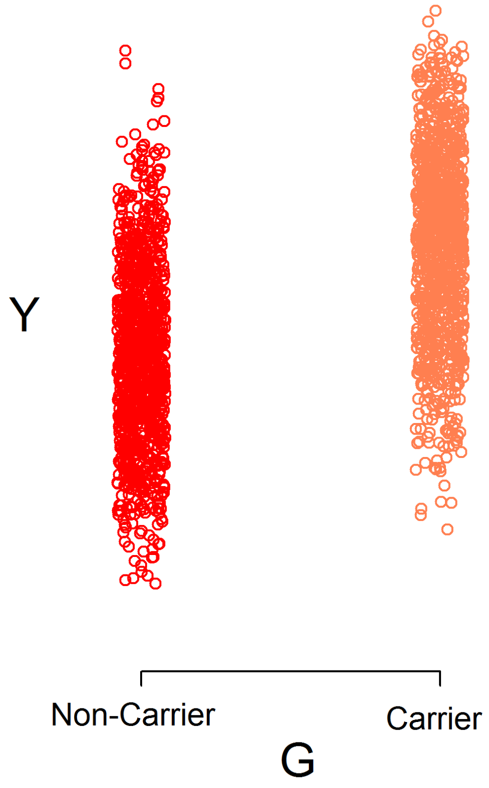
- Use of common regression and testing commands, in simple genetic settings
- Some useful post-processing commands, after the regression is done

NB Because regression is a vast subject, the help files for commands in this session are also vast. If you are new to regression, Dalgaard's book is a good place to look for more material.

Comparing means: two groups

Simple data (below) suggests a simple model;

- All outcomes (Y) independent, i.e. one from each person in your study
- Within each group (defined by G) there is a mean outcome
- ... are the means different?



The t-test is the standard statistical tool for making this comparison. Common to recode G (carrier/non-carrier) as 1/0 – and to call it X, or covariate/predictor/dependent variable.

Comparing means: two groups

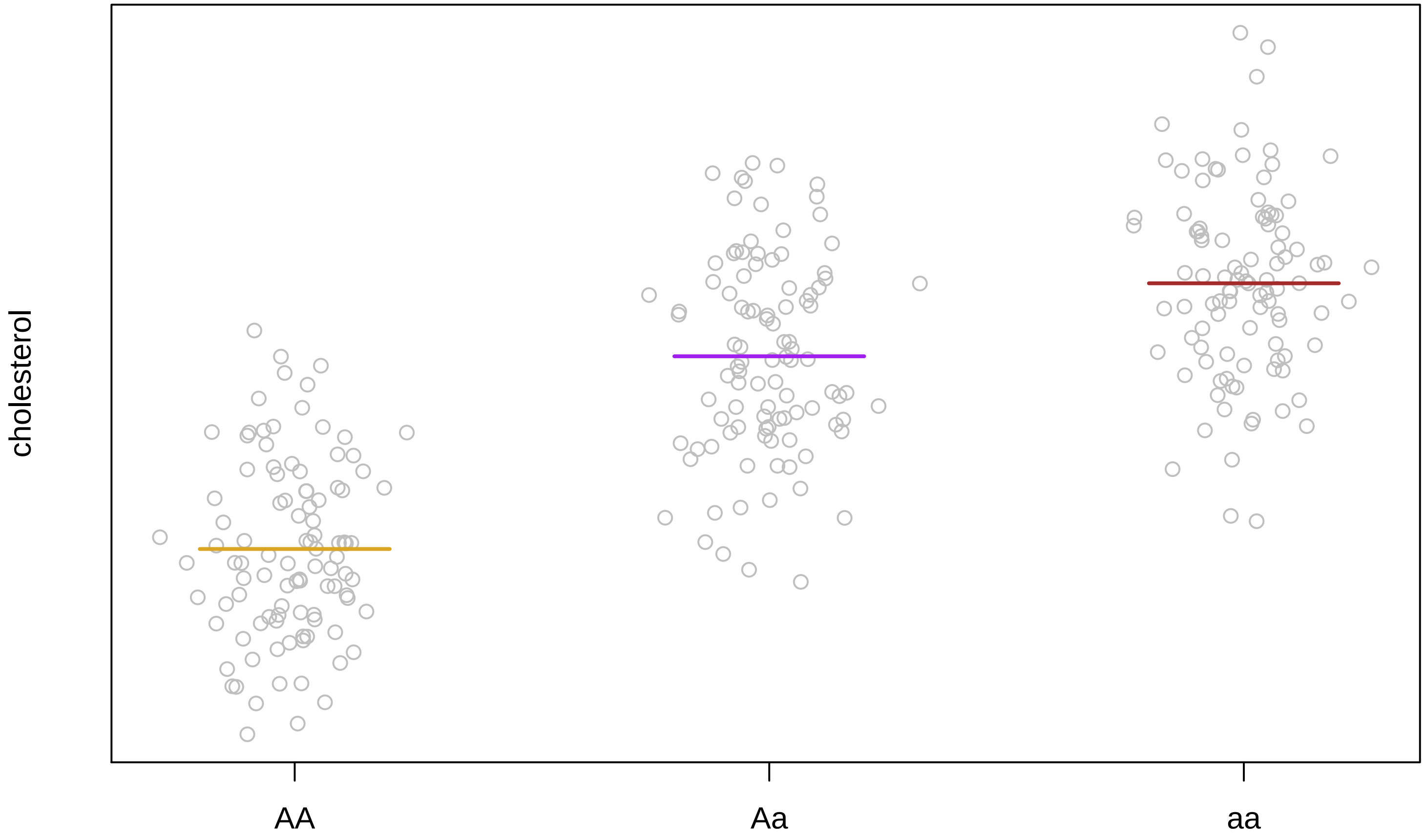
Straightforward R command to do this;

```
> t.test(y˜g, data=mydata)

        Welch Two Sample t-test

data:  y by g
t = 2.4841, df = 995.723, p-value = 0.01315
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.2320599 1.9775503
sample estimates:
    mean in group carrier mean in group non-carrier
                1.5572602                 0.4524551
```

- $Y \sim X$ formula, just as in graphics
- Confidence interval is for difference in means (1st - 2nd)
- p-value is two-sided (see `alternative`) and does not assume equal variances (see `var.equal`)
- Also accepts vector input, for one-sample & paired tests

Comparing means: multiple groups

In a new study, with more groups, how do the means compare?



... need to make/combine two comparisons, here

Comparing means: multiple groups

Assuming we have genotypes G coded ”AA”/”Aa”/”aa”;

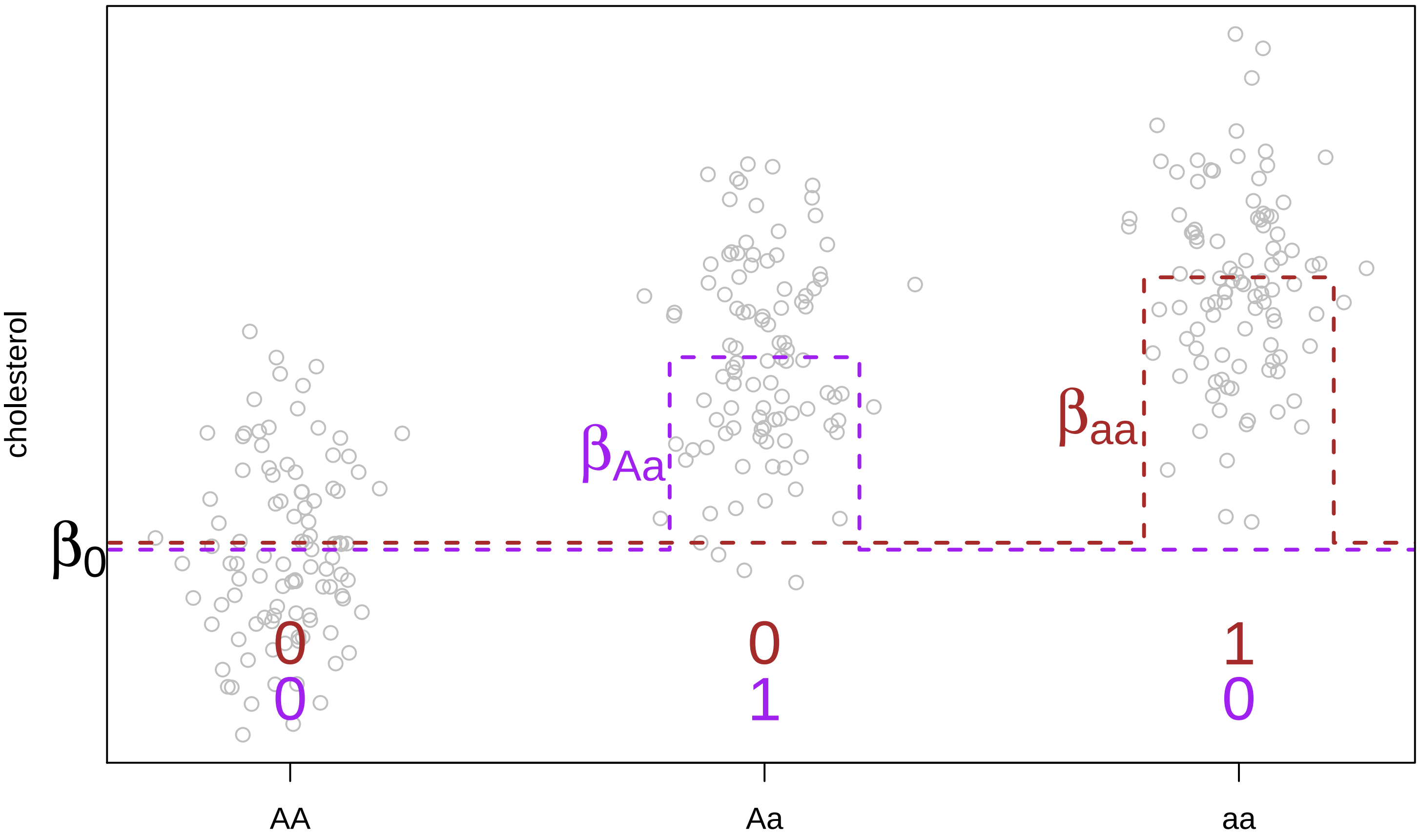
```
> aov1 <- aov( chol ~ g, data=mynewdata )

> aov1
Call:
   aov(formula = chol ~ g)
Terms:
                      gg Residuals
Sum of Squares  193.0185  135.4168
Deg. of Freedom        2       297
Residual standard error: 0.6752397
Estimated effects may be unbalanced

> summary(aov1)
             Df Sum Sq Mean Sq F value Pr(>F)
gg            2  193.0   96.51   211.7 <2e-16 ***
Residuals   297  135.4    0.46
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

This is *Analysis of Variance*. Use `model.tables(aov1, type="means")` to see and compare the means (!)

Comparing means: multiple groups

A more direct way to say the same thing;

$$\text{Mean}(Y) = \beta_0 + \beta_{Aa} \times (G == Aa) + \beta_{aa} \times (G == aa)$$



Comparing means: multiple groups

With genotypes stored as a factor, we can perform the inference using `lm()` – for Linear Model;

```
> lm1 <- lm(chol~g, data=mynewdata)
> summary(lm1)
Call:
lm(formula = chol ~ g)
Residuals:
     Min       1Q   Median       3Q      Max
-1.70228 -0.48623 -0.02692  0.47186  1.79321
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.06757    0.06752   1.001    0.318
gAa         1.37916    0.09549  14.442   <2e-16 ***
gaa         1.90149    0.09549  19.912   <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.6752 on 297 degrees of freedom
Multiple R-squared:  0.5877,    Adjusted R-squared:  0.5849
F-statistic: 211.7 on 2 and 297 DF,  p-value: < 2.2e-16
```

Comparing means: multiple groups

Notes on this fairly verbose `summary()`;

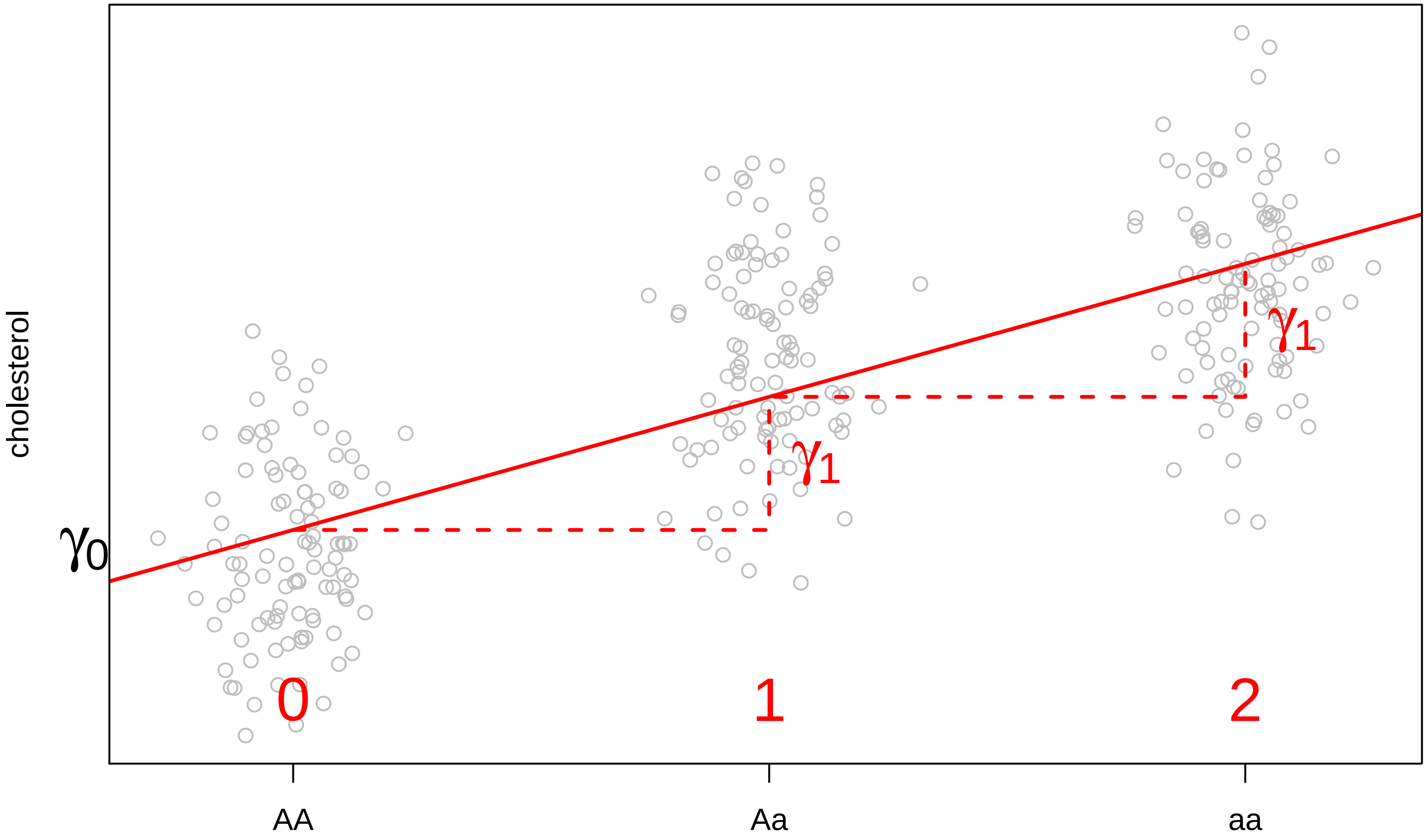
- `lm()` takes formula input
- Get same F statistic as analysis of variance – doing the same analysis, comparing means
- 'Wald tests' of individual coefficients also given; is the intercept zero, is the difference between mean Y in Aa and AA zero?
- Alpha-numerically 'first' level of factor is chosen as reference – unless you specify otherwise, when making a `factor()`. Or `relevel()` an existing factor
- This analysis assumes variance of outcomes *is* constant across the groups – slightly different to the default t-test.

Turn those #&%ing stars off with `options(show.signif.stars=FALSE)`

Comparing means: multiple groups

A more common use of `lm()`;

$$\text{Mean}(Y) = \gamma_0 + \gamma_1 \times \#\text{minor alleles}$$



Comparing means: multiple groups

The work here is constructing the 0/1/2 covariate; one approach (below) exploits R's 'coercion' of TRUE/FALSE to 1/0, in math expressions;

```
> mynewdata$g.num <- with(mynewdata, 0 + 1*(g=="Aa") + 2*(g=="aa"))
> lm2 <- lm(chol~g.num, data=mynewdata)
> summary(lm2)
Call:
lm(formula = chol ~ g.num)

Residuals:
     Min       1Q   Median       3Q      Max
-1.84509 -0.47012 -0.08037  0.52075  1.66926

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.21037    0.06426   3.274  0.00119
g.num        0.95074    0.04977  19.101  < 2e-16

Residual standard error: 0.7039 on 298 degrees of freedom
Multiple R-squared:  0.5504,    Adjusted R-squared:  0.5489
F-statistic: 364.9 on 1 and 298 DF,  p-value: < 2.2e-16
```

Comparing means: multiple groups

As well as the point estimates and p-values, we probably want confidence intervals for the parameters;

```
> confint(lm2)
                 2.5 %    97.5 %
(Intercept) 0.08391672 0.3368275
g.num       0.85279147 1.0486953
```

For tests that do not require constant variance – but that do require large sample sizes;

```
> library("sandwich")
> library("lmtest")
> waldtest(lm2, "g.num", vcov=vcovHC(lm2) )
Wald test

Model 1: chol ~ g.num
Model 2: chol ~ 1
  Res.Df Df      F    Pr(>F)
1    298
2    299 -1 396.59 < 2.2e-16
```

Comparing means: multiple groups

It is possible to extract most of what you need from `lm2` or `summary(lm2)` using the $ syntax. But it's easier to use extractor functions;

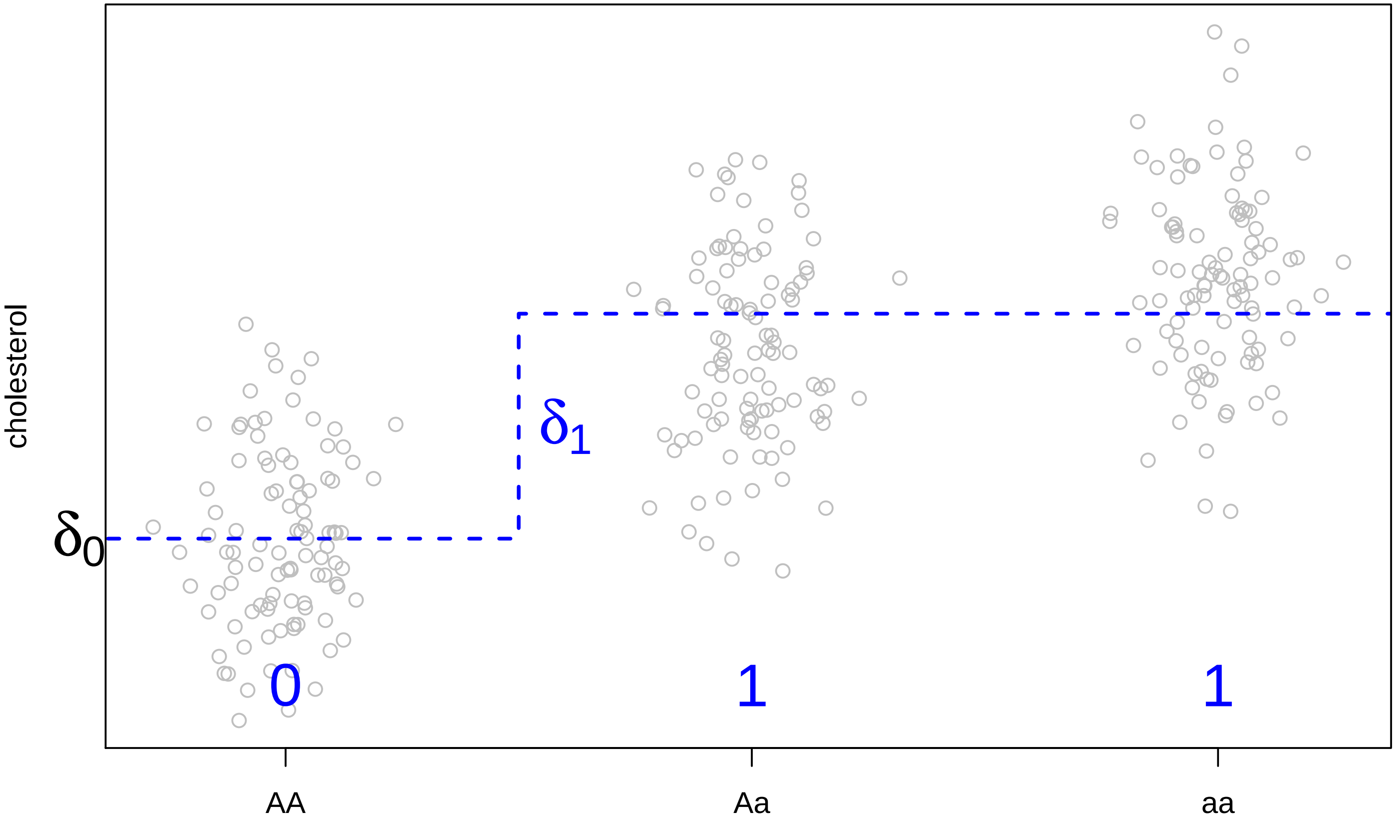
- `coef()`; the estimated coefficients
- `predict()`; predicted values at given covariates
- `fitted.values()`; fitted values for original data
- `residuals()`; residuals for original data
- `confint()`; see earlier slides
- `vcov()`; variance-covariance matrix for the point estimates
- `vcovHC()`; robust version – in the `sandwich` package
- `AIC()`, `BIC()`; An Information Criterion (and another one)

These can also be used on output from other regression functions. See also `?influence.measures` for diagnostic tools.

Comparing means: multiple groups

To fit the 'dominant model';

$$\text{Mean}(Y) = \delta_0 + \delta_1 \times (G \neq AA)$$

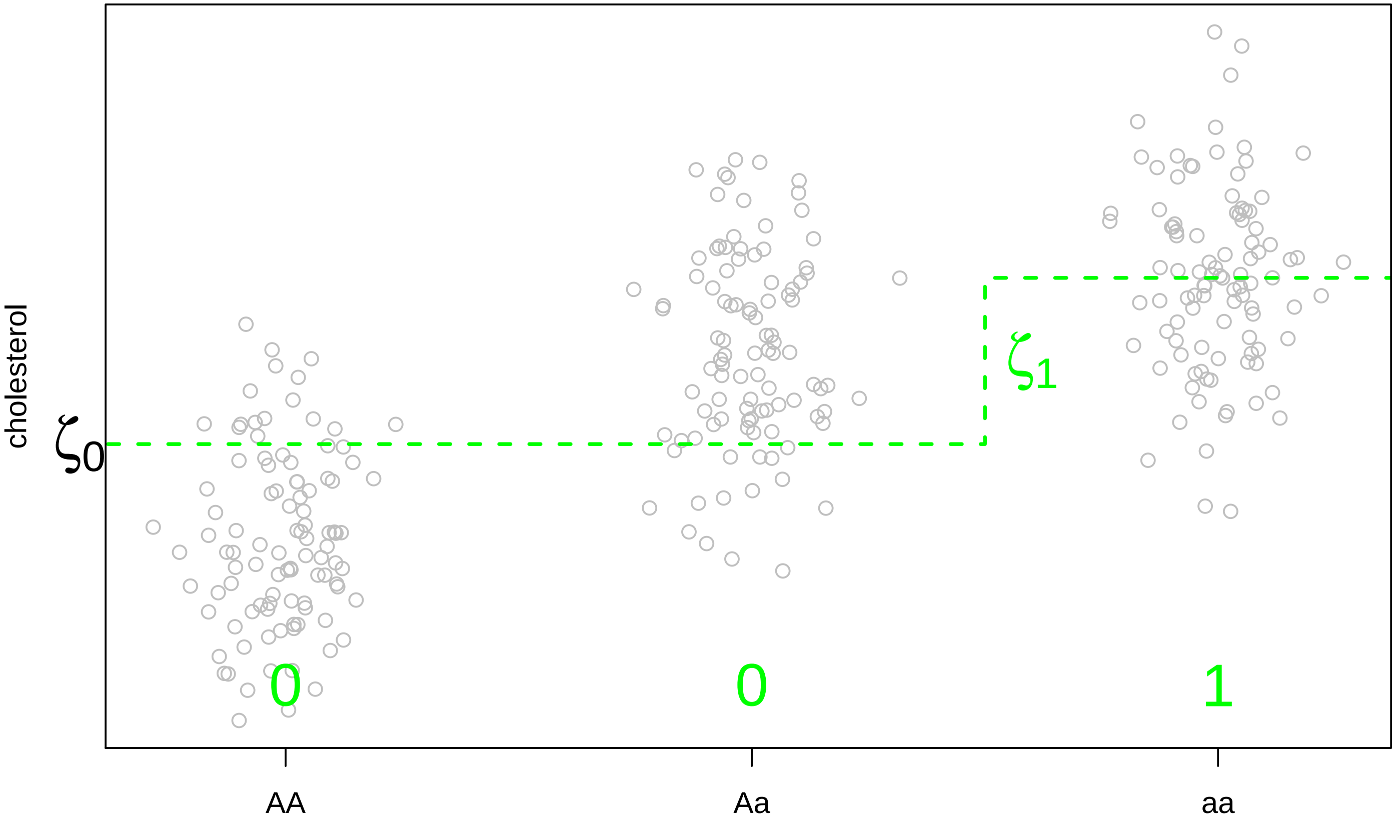


...define `g.num2 <- g!="AA"` and regress `Y~g.num2`.

Comparing means: multiple groups

To fit the ‘recessive model’;

$$\text{Mean}(Y) = \zeta_0 + \zeta_1 \times (G == AA)$$



…define `g.num3 <- g=="AA"` and regress `Y`∼`g.num3`.

Notes

- There are many ways to convert stored genotypes to the 0/1/2 variables R uses in regression

- Check *you* got it right, before doing regression. Use e.g. `table()` to ensure everything matches up

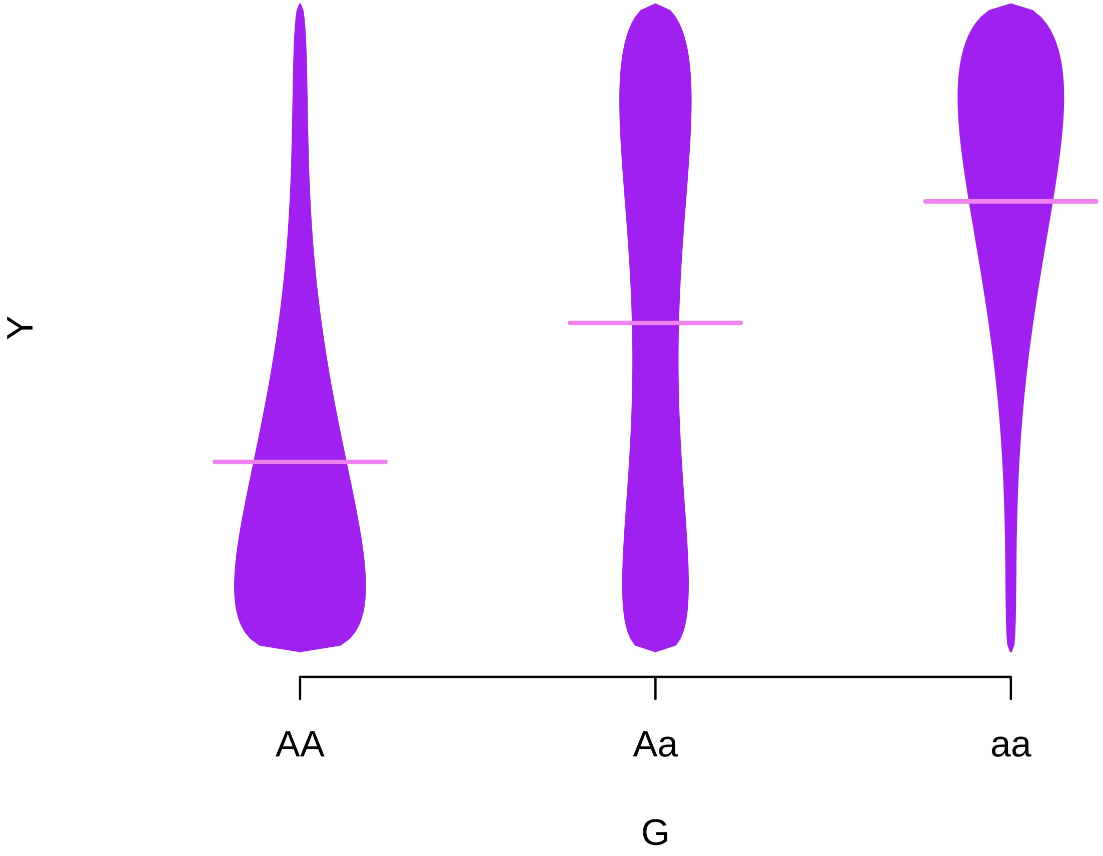
- Regressing on a factor, R actually sets up multiple binary covariates, and regresses on each of them

- When missing values are present in outcome or any covariates in the formula, R drops that row of the data before starting analysis

- Intercepts are implicit in R regression formula. Should you need to, take them out with e.g. `Y~ -1 + g`. See `?formula` for more tricks like this

Comparing means: adjustments

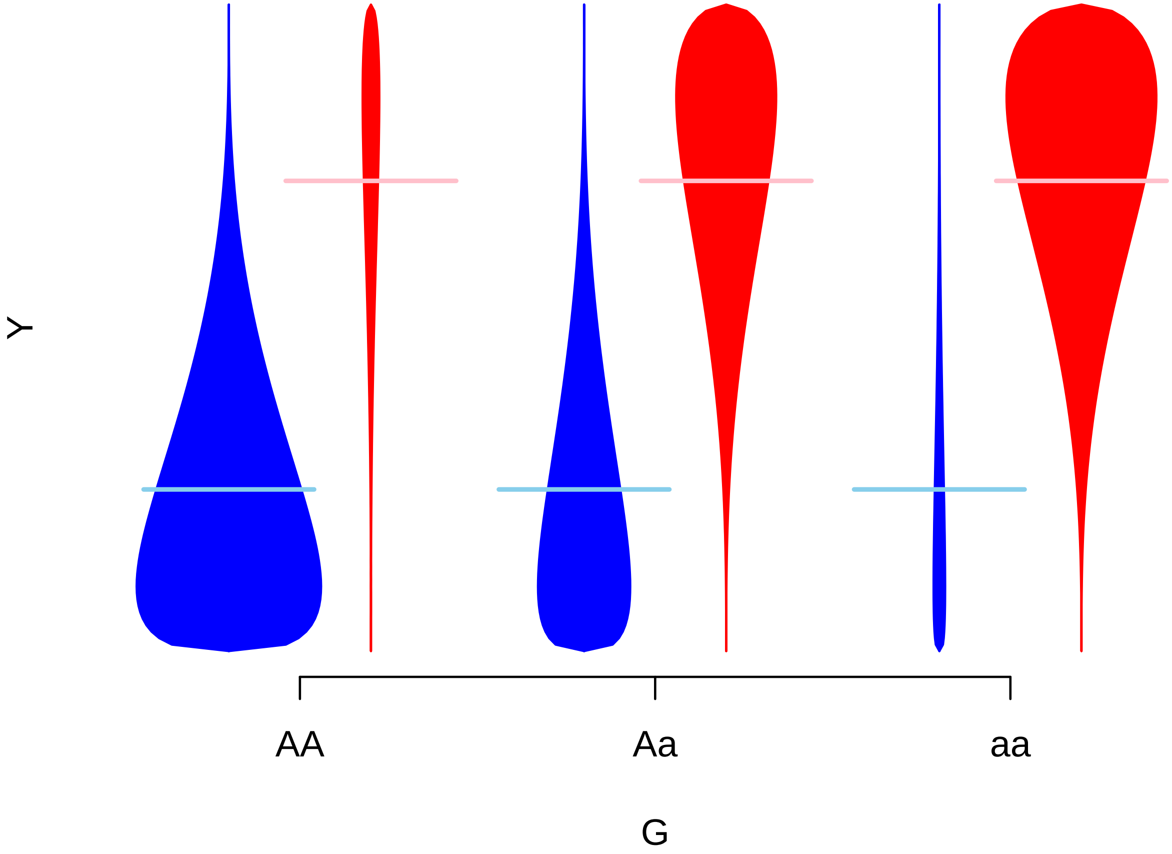
Imagine, in a huge sample, we see association between phenotype and genotype;



... `lm(y~g)` would report a positive slope, very unlikely by chance alone – and this is statistically ‘right’.

Comparing means: adjustments

But this is *scientifically* unimpressive, if breaking the same data down by ancestry group we see this;



The effect is known as *population stratification* – statisticians know it as *confounding*.

Comparing means: adjustments

To fit models where

$$\text{Mean}(Y) = \beta_0 + \beta_1 \times G + \beta_2 \times Z$$

the R formula syntax is

$$\mathrm{y} \sim \mathrm{g} + \mathrm{z}$$

... so on the previous slide, with Z = (colour==red), the `g` coefficient (β_1) tells us about how the Mean(Y) varies with G in both red and blue populations; should fit $\beta_1 \approx 0$, here.

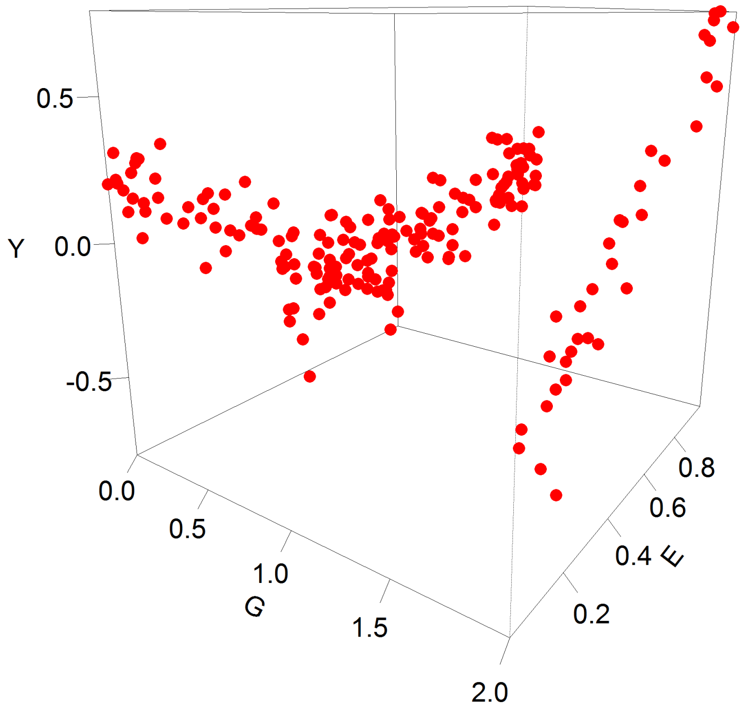
To adjust for multiple covariates (e.g. age & sex & different principal components of genotype data, representing ancestry);

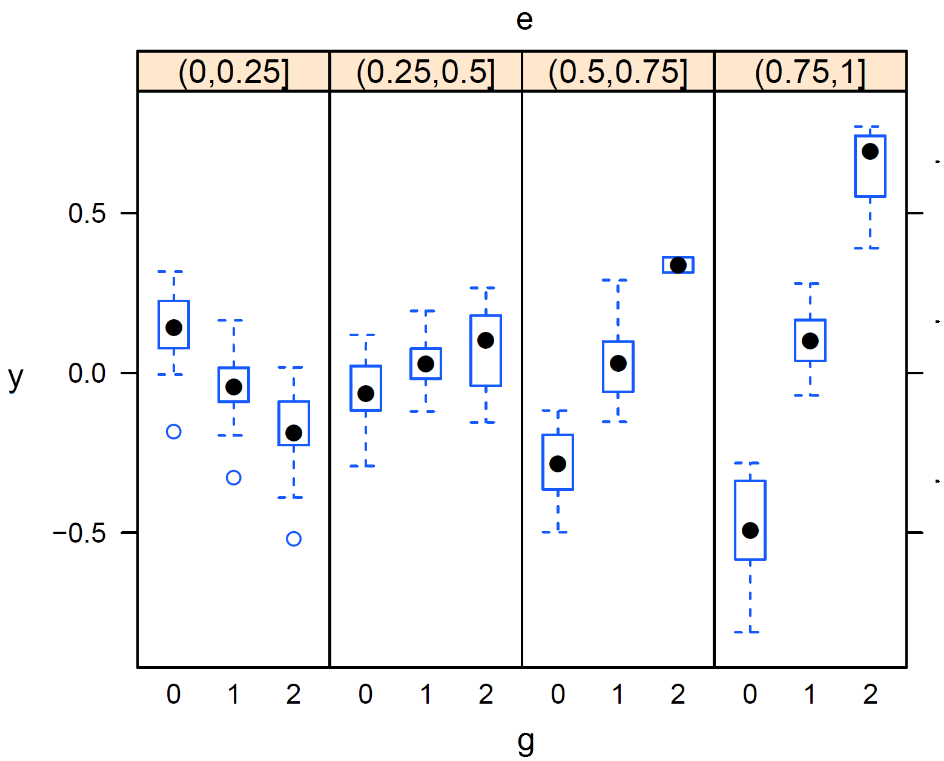
$$\mathrm{y} \sim \mathrm{g} + \mathrm{age} + \mathrm{sex} + \mathrm{pc1} + \mathrm{pc2} + \mathrm{pc3} + \mathrm{pc4} + \mathrm{pc5}$$

Note PCs can be obtained with `princomp()` and/or `prcomp()`.

Comparing means: interaction

When we have data on genotype (G=0/1/2) and environment (0≤E≤1) – here presented in two sub-optimal ways;

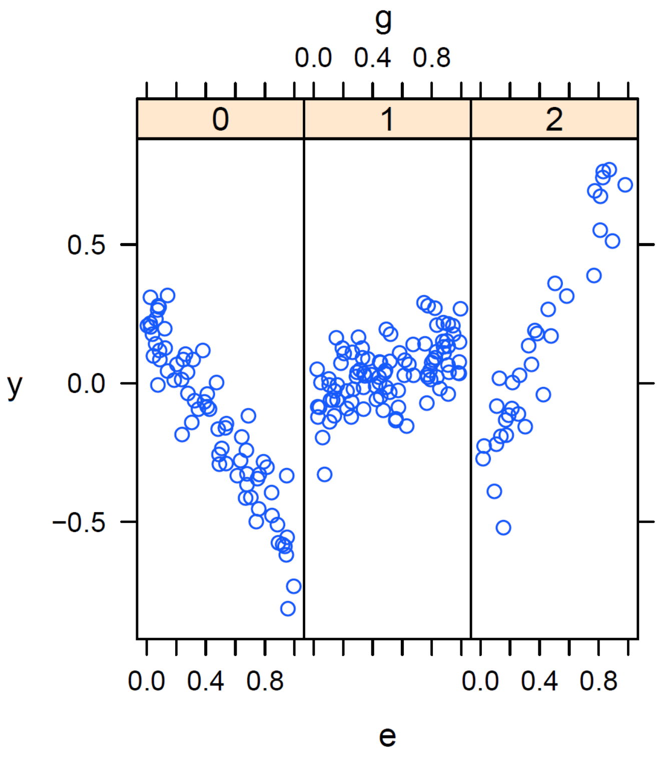


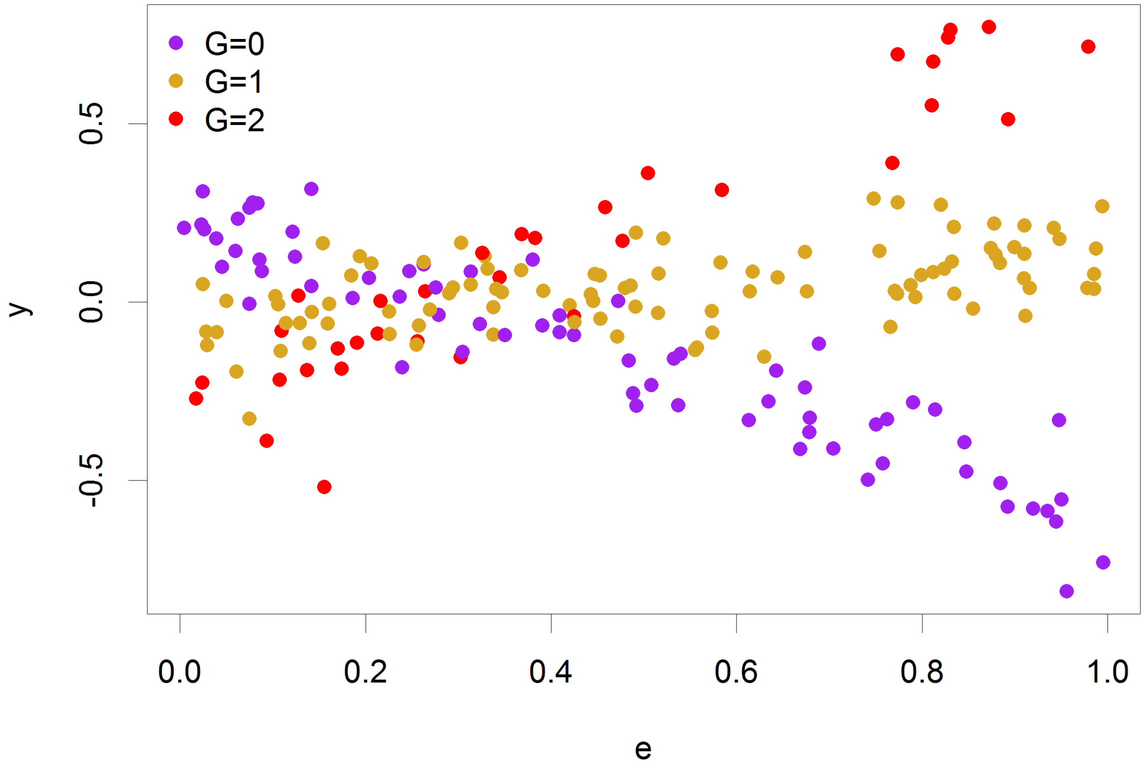


(These results using `persp()` and `bwplot()`)

Comparing means: interaction

And two simpler ways; (using `xyplot()` and `plot()`)





Does the slope of the $Y - E$ relationship differ according to G?

Comparing means: interaction

With e.g. G=number of minor alleles, we might fit

$$\text{Mean}(Y) = \beta_0 + \beta_1 G + \beta_2 E + \beta_3 G \times E$$

In R this is achieved by

```
> lm3 <- lm(y~ g + e + g:e, data=mylastdata)
> summary(lm3)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.23908    0.01999   11.96   <2e-16
g           -0.29304    0.01814  -16.16   <2e-16
e           -0.83177    0.03633  -22.89   <2e-16
g:e          1.01037    0.03431   29.45   <2e-16
```

- In formulas, colons (:) denote interaction
- Shorthand `y∼g*e` denotes interactions *and* all main effects
- For math in formulas, can use `I()` to *insulate*, for example `y∼ g*I(sbp-dbp)`, for interactions with pulse pressure

Comparing odds

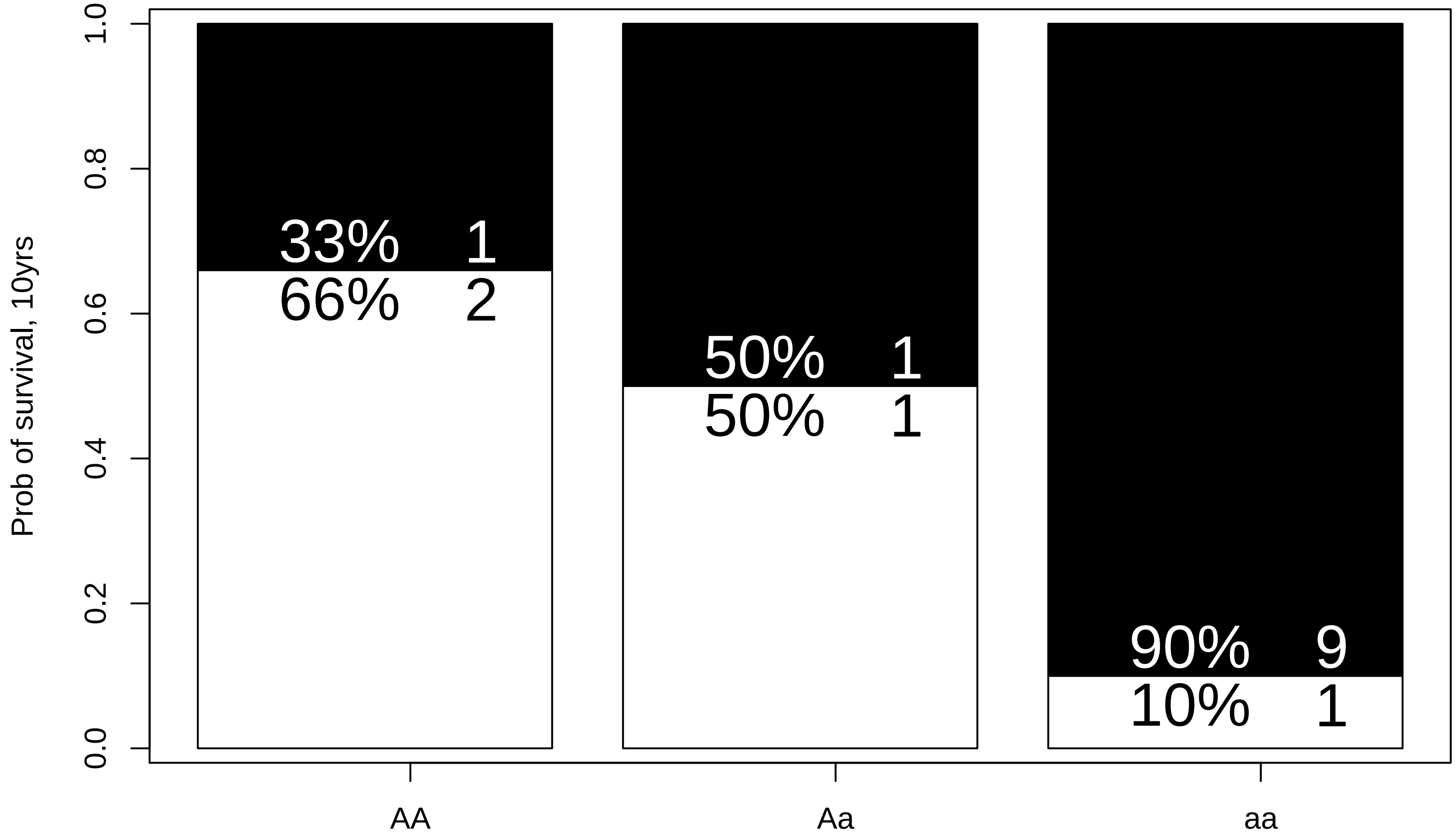
Logistic regression is the ‘default’ analysis for binary outcomes

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

What are odds? Really just probability...

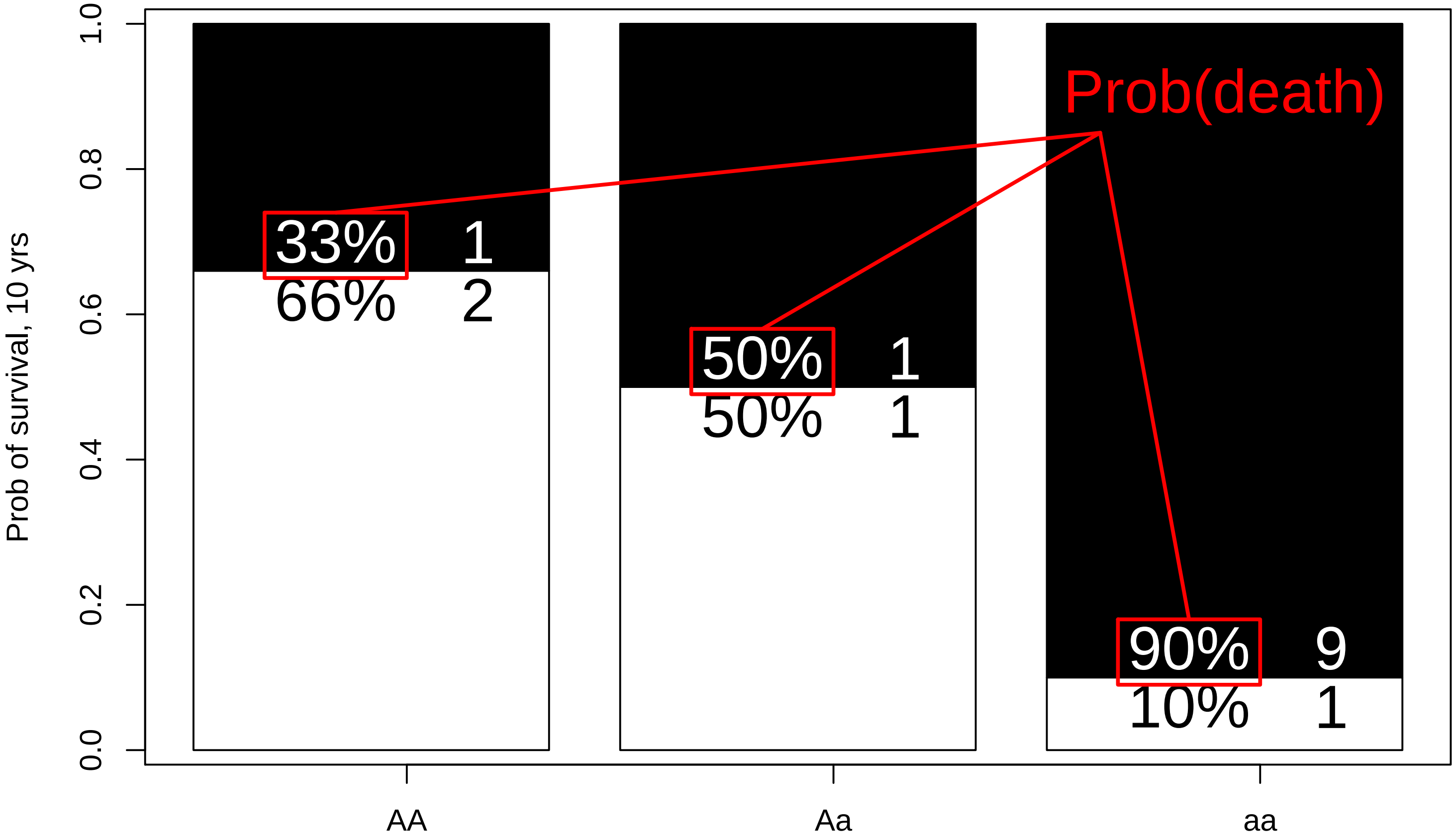
Comparing odds: what are odds?

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



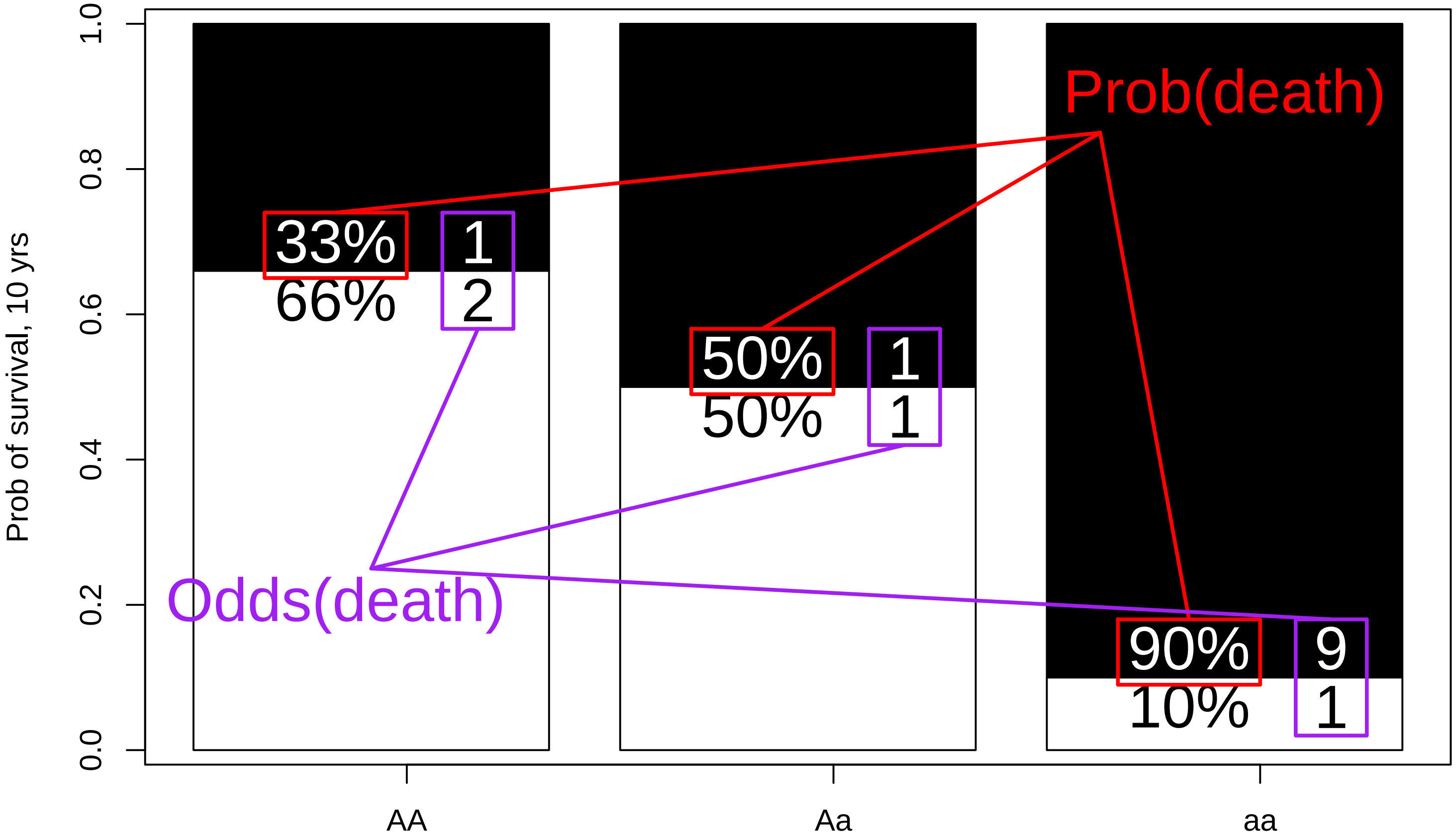
Comparing odds: what are odds?

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



Comparing odds: what are odds?

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



– so what are **odds ratios**?

Comparing odds: what are odds?

Using the data from the previous slide, with g stored as a factor, levels "AA"/"Aa"/"aa";

```
> glm1 <- glm( dead10yrs ~g, family=binomial, data=myposthocdata)
> coef(glm1)
(Intercept)         gAa         gaa
 -0.6931472   0.6931472   2.8903717
```

- First term is estimate of *log odds* in reference group (AA) – to transform to an estimate of odds, use $e^{-0.6931}$=0.5
- Other terms are estimates of *log odds ratios*, relative to the reference group; to transform to OR, use `exp()` to obtain $e^{0.6931} = 2, e^{2.8904} = 18$
- If/when you forget the `family=binomial` argument, default is linear regression, also given by `lm()`

Comparing odds: what are odds?

Confidence intervals and p-values are obtained as with `lm()` output – as here for the log odds ratios;

```
> confint(glm1)
Waiting for profiling to be done...
                2.5 %    97.5 %
gAa         0.1242838  1.2723849
gaa         2.1529671  3.7154673
> confint.default(glm1)
                2.5 %    97.5 %
gAa         0.1201986  1.2660957
gaa         2.1148912  3.6658523
> summary(glm1)
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
gAa           0.6931     0.2923   2.371  0.01773
gaa           2.8904     0.3957   7.305 2.77e-13
```

- Most users expect the `confint.default()` intervals
- Use `exp()` on `confint()` output (either version) to get intervals for the corresponding odds ratios.

Other model-fitting commands

For an inclusive definition of 'model';

- `fisher.test()` and `chisq.test()` perform Fisher's exact test and Pearson's χ^2 test, on contingency tables
- `coxph()` in the `survival` package, for Cox Proportional Hazards regression
- `gee()` in the `gee` package, for Generalized Estimating Equations
- `lmer()` and `glmer()` in the `lme4` package fit (Generalized) Linear Mixed Models
- `ns()` and `bs()` in the `splines` package calculate natural and B-splines

Search the R/Bioconductor sites to see how to fit many other models.