Introduction to R for Data Analysis in the Health Sciences: Lecture 6

Amy Willis, Biostatistics, UW

15 November, 2019

Today

- Linear regression
  - Interactions
  - Intercept-free
- Logistic regression
  - Fitting
  - Interpretation

As always. . .

```
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------
```

```
## v ggplot2 3.1.0     v purrr   0.3.0
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.3.1
## v readr   1.1.1     v forcats 0.2.0
```

```
## -- Conflicts ------------------------------------ tidyv
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

Reading in the data

Let's look again at the FEV data set that we have analysed previously:

```
fev <- read_csv("datasets/fev.csv")
```

```
## Parsed with column specification:
## cols(
##   seqnbr = col_integer(),
##   subjid = col_integer(),
##   age = col_integer(),
##   fev = col_double(),
##   height = col_double(),
##   sex = col_integer(),
##   smoke = col_integer()
## )
```

Clarifying the data

We began by clarifying how variables are coded:

```
fev_char <- fev %>%
  mutate(sex = ifelse(sex == 1, "male", "female"),
         smoke = ifelse(smoke == 1, "smoker", "nonsmoker"))
```

Fitting interactions

We are often interested in fitting a model where variables "interact"

- The value of the coefficient of one variable depends on the coefficient of another variable

e.g. The model for how age increases the expected value of FEV depends on sex:

- $E(FEV_i) = \beta_{0f} + \beta_{1f} \times age_i$ for female children
- $E(FEV_i) = \beta_{1m} + \beta_{1m} \times age_i$ for male children

Here we have a sex-specific intercept, and a sex-specific slope

Fitting interactions

We fit this model as follows:

```
lm_i <- lm(fev ~ sex * age, data = fev_char)
```

- `fev ~ sex * age` fits an interactive model
  - A different linear model for `fev` on age for each `sex`
- `fev ~ sex + age` fits an additive model
  - A linear model for `fev` on age, with a different intercept for each `sex`

Fitting interactions

The output format is similar to before, but now intercept and slope terms indicate if they apply only for male subjects:

```
lm_i
```

```
##
## Call:
## lm(formula = fev ~ sex * age, data = fev_char)
##
## Coefficients:
## (Intercept)      sexmale          age  sexmale:age
##      0.8495      -0.7759       0.1627       0.1107
```

Fitting interactions

The fitted values from this model are

- $\hat{FEV}_i = 0.849 + 0.163 \times age_i$ for female children
- $\hat{FEV}_i = (0.849 - 0.776) + (0.163 + 0.111) \times age_i$ for male children

Add the values from the baseline category to the non-baseline category to get non-baseline category coefficient estimates.

Fitting interactions

Note this fits (almost*) the same model as if we fit each sex individually:

```
lm(fev ~ age,
   data = fev_char %>% filter(sex == "female")) %>% coef
```

```
## (Intercept)         age
##   0.8494671   0.1627289
```

```
lm(fev ~ age,
   data = fev_char %>% filter(sex == "male")) %>% coef
```

```
## (Intercept)         age
##  0.07360056  0.27347763
```

*Almost, because the same variance is estimated for both groups in the interactive model, while different variances are estimated for the separate models.

Fitting interactions

We can get more information with `summary`

```
lm_i %>% summary
```

```
##
## Call:
## lm(formula = fev ~ sex * age, data = fev_char)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.64072 -0.34337 -0.04934  0.33206  1.86867
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.849467   0.102199   8.312 5.51e-16 ***
## sexmale     -0.775867   0.142745  -5.435 7.74e-08 ***
## age          0.162729   0.009952  16.351  < 2e-16 ***
## sexmale:age  0.110749   0.013786   8.033 4.47e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5196 on 650 degrees of freedom
```

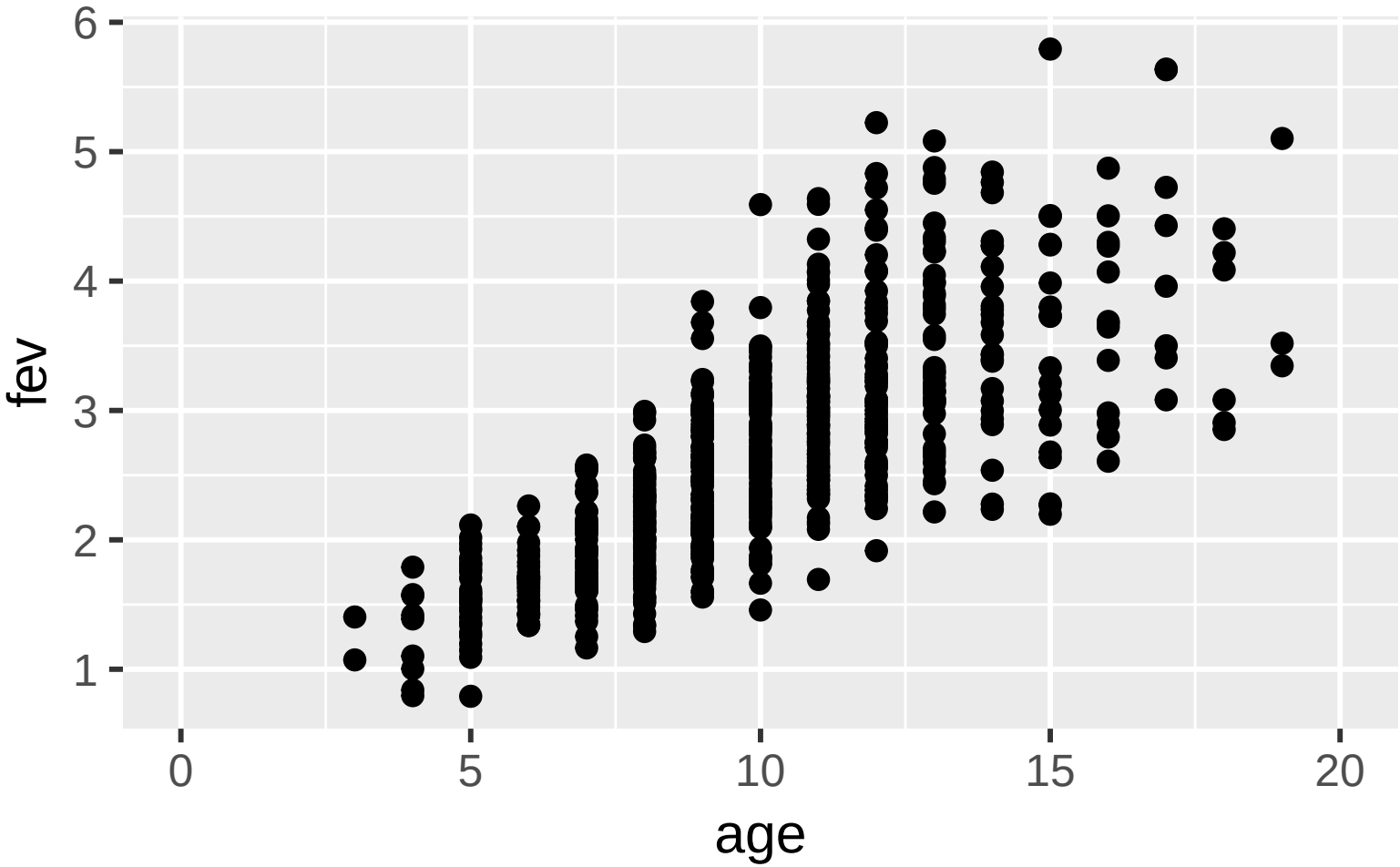
Fitting interactions

Questions?

Linear models with & without an intercept

What do we expect the FEV of a 0-year-old infant to be?

```
fev_char %>% ggplot(aes(x = age, y = fev)) +
  geom_point() + xlim(0, 20)
```



Linear models with & without an intercept

Recall that, by default, `lm` includes an intercept in the model:

```
lm(fev ~ age, data = fev_char)
```

```
##
## Call:
## lm(formula = fev ~ age, data = fev_char)
##
## Coefficients:
## (Intercept)          age
##      0.4316       0.2220
```

Linear models with & without an intercept

It is uncommon, but sometimes we want to fit a model without an intercept:

$$E(FEV_i) = \beta \times age_i$$

We can do this as follows:

```
lm(fev ~ age - 1, data = fev_char)
```

```
##
## Call:
## lm(formula = fev ~ age - 1, data = fev_char)
##
## Coefficients:
##   age
## 0.262
```

Linear models with `lm`

Any model fit with `lm` produces the same type of output

- Additive, interactive, intercept-free

You can use `coef`, `summary`, `anova` on any `lm` output

Linear models with & without an intercept

Questions?

Other regression-type models

The linear model is a common choice when we have a continuous response (Y) variable.

What happens when we have a *binary* response variable?

Models for binary data

Binary/dichotomous variables can be coded as 0 or 1. A value of 1 indicates which option occurred – hence these are sometimes called *indicator* variables.

Common examples include

- Mortality: 0 = alive; 1 = dead
- Remission: 0 = no; 1 = yes
- Sex: 0 = male; 1 = female
- Intervention: 0 = control; 1 = treated

Logistic regression: predicting malignancy

We're going to look at a dataset containing information about malignant and benign breast cancers:

```
bc <- read_csv("datasets/breastcancer.csv")
```

```
## Parsed with column specification:
## cols(
##   Id = col_integer(),
##   Cl.thickness = col_integer(),
##   Cell.size = col_integer(),
##   Cell.shape = col_integer(),
##   Marg.adhesion = col_integer(),
##   Epith.c.size = col_integer(),
##   Bare.nuclei = col_integer(),
##   Bl.cromatin = col_integer(),
##   Normal.nucleoli = col_integer(),
##   Mitoses = col_integer(),
##   Class = col_character()
## )
```

Logistic regression: predicting malignancy

This dataset is from a clinician at UW Madison (for details, see the package `mlbench`). Each row describes a cancer presented to the clinician with respect to

- categorical – unordered: Sample ID
- categorical – ordered 0 to 10: Clump Thickness, Uniformity of Cell Size, Uniformity of Cell Shape, Marginal Adhesion, Single Epithelial Cell Size, Bare Nuclei, Bland Chromatin, Normal Nucleoli, Mitoses
- categorical – unordered (benign or malignant): Class

Logistic regression: predicting malignancy

Let's take a quick look at the dataset:

```
bc %>% names
```

```
##  [1] "Id"              "Cl.thickness"    "Cell.size"
##  [4] "Cell.shape"      "Marg.adhesion"   "Epith.c.size"
##  [7] "Bare.nuclei"     "Bl.cromatin"     "Normal.nucleol
## [10] "Mitoses"         "Class"
```

```
bc$Class %>% unique
```

```
## [1] "benign"    "malignant"
```

```
bc$Cell.size %>% unique %>% sort
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

Logistic regression: predicting malignancy

The response variable that we want to model is whether the cancer is malignant. We create an indicator variable Y that takes value 0 when the cancer is benign and 1 if the cancer is malignant.

```
bc_y <- bc %>%
  mutate(y = ifelse(Class == "benign", 0, 1))
```

Logistic regression: predicting malignancy

Logistic regression fits the model

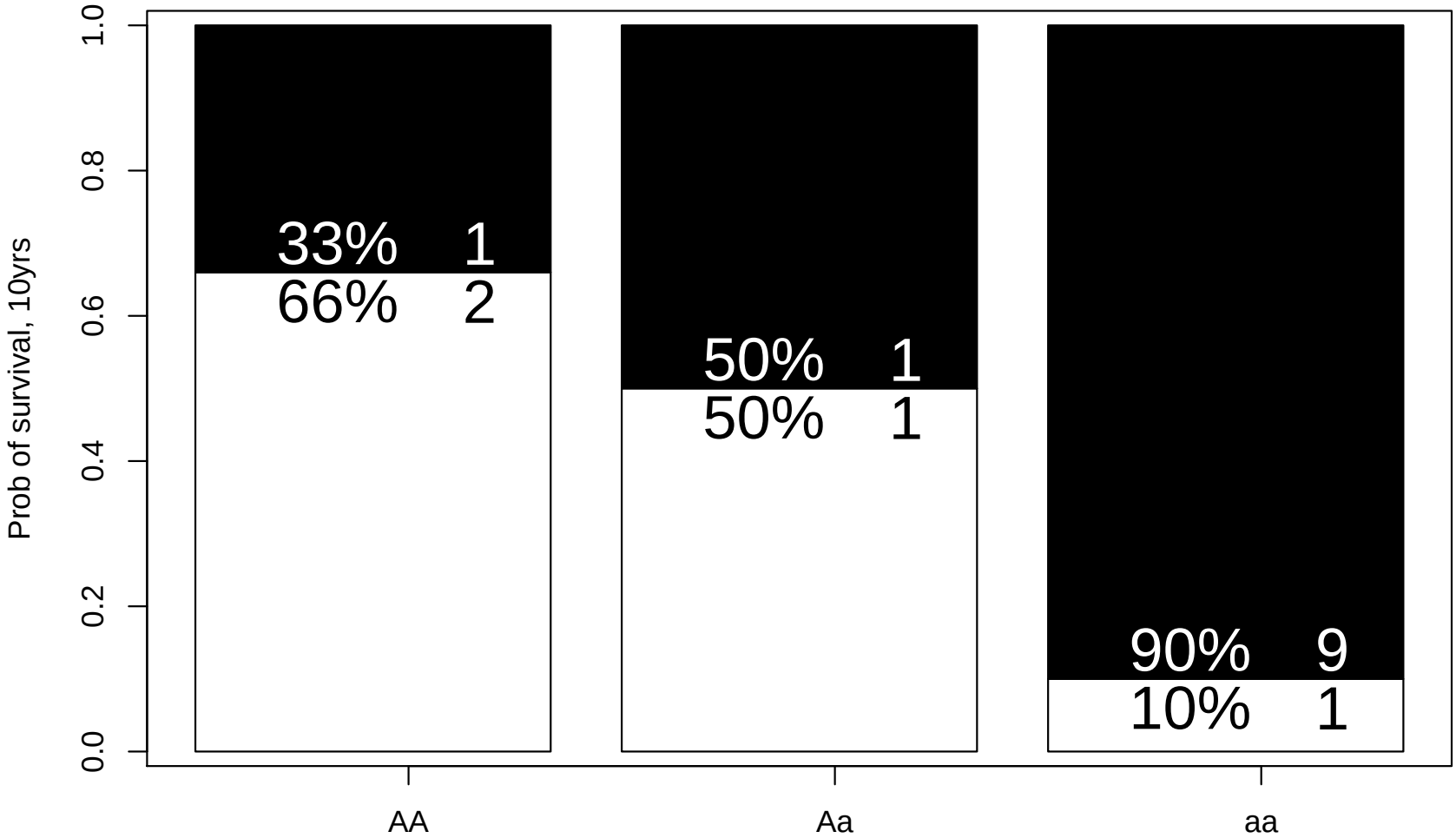
$$\log\left(\frac{Pr(Y_i = 1|X_i = x_i)}{Pr(Y_i = 0|X_i = x_i)}\right) \quad = \quad \beta_0 + \beta_1 x_i$$

It is linear model for the logarithm of the *odds* that $Y = 1$.

How can we interpret odds?

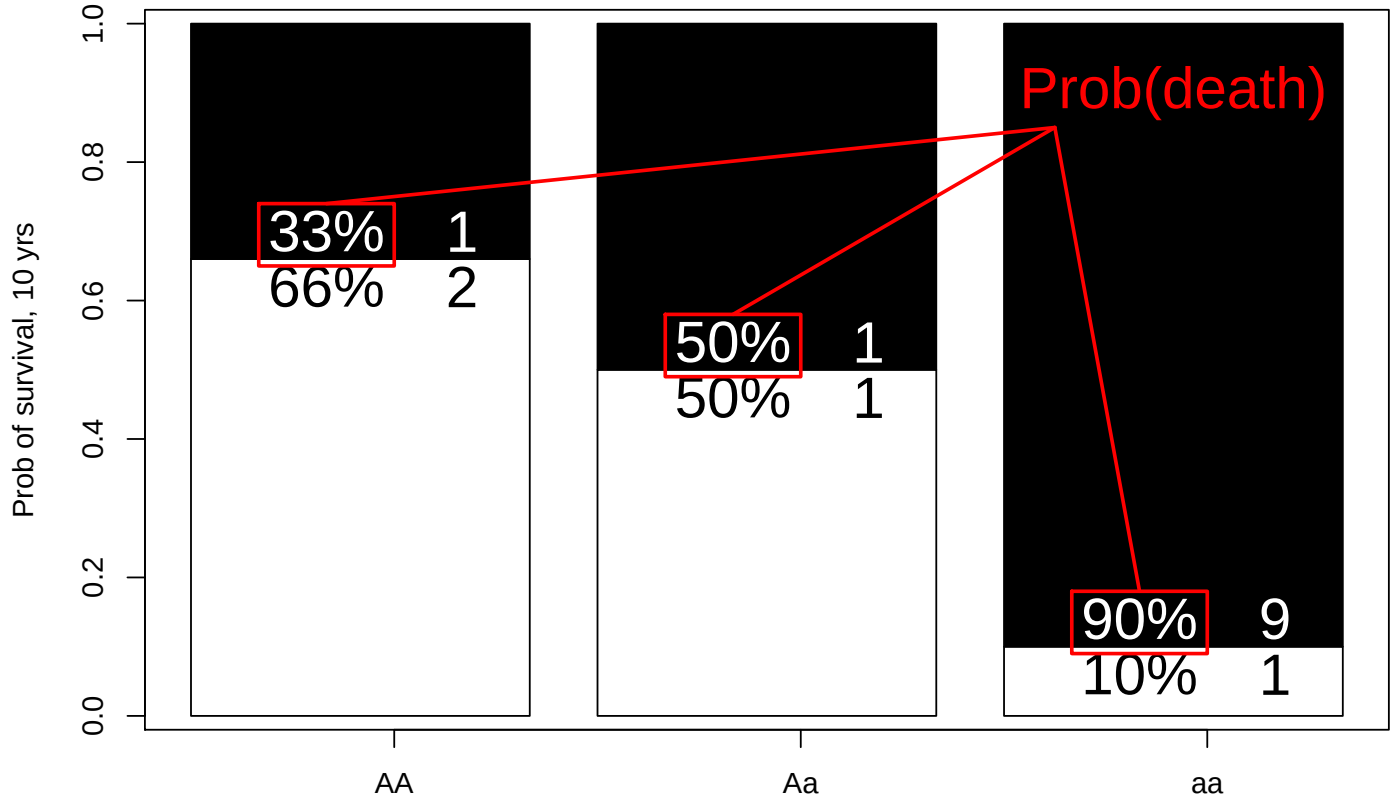
Interpreting odds

Suppose that patients' survival after diagnosis depends on cancer subgroup (AA, Aa, or aa). Patients survival after 1 year is given below:



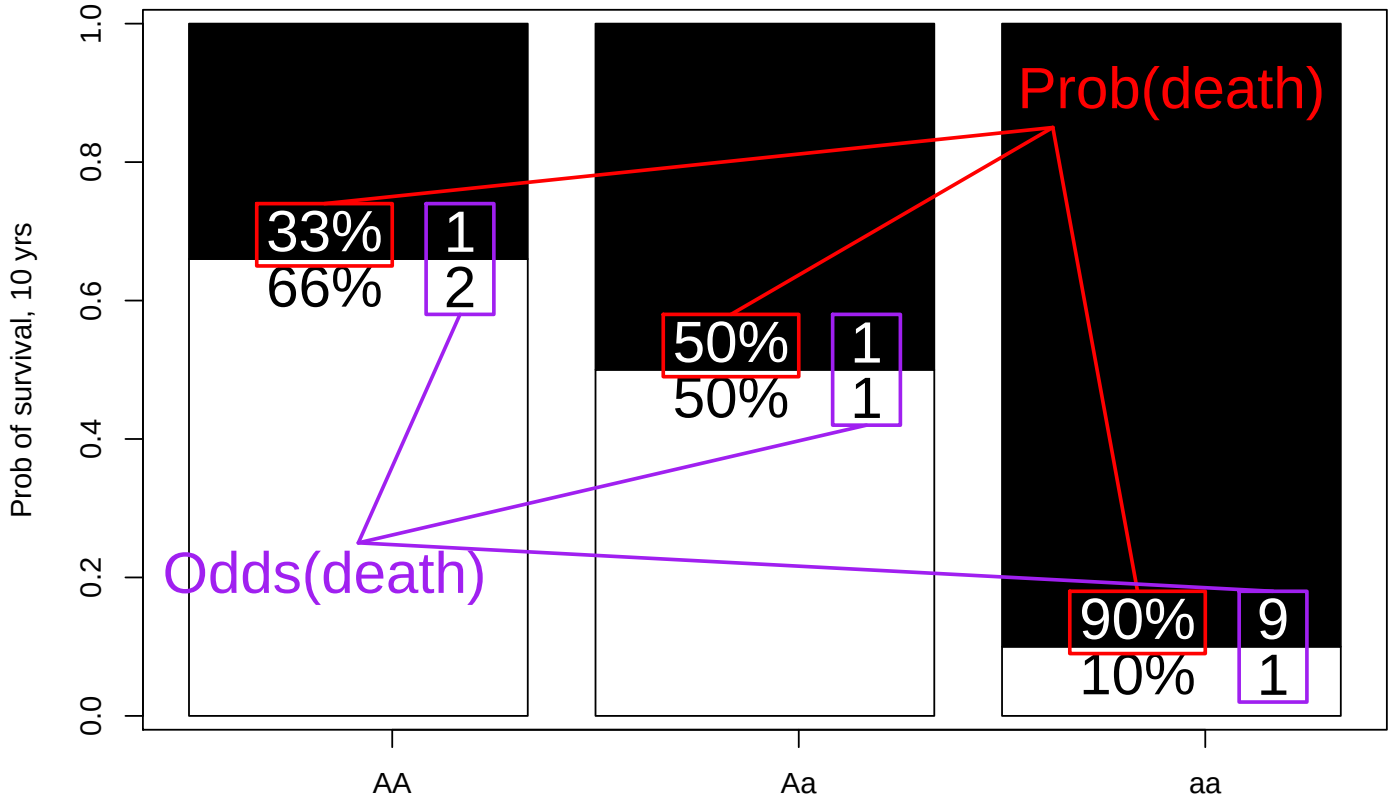
Interpreting odds

What is the probability of death for each group?



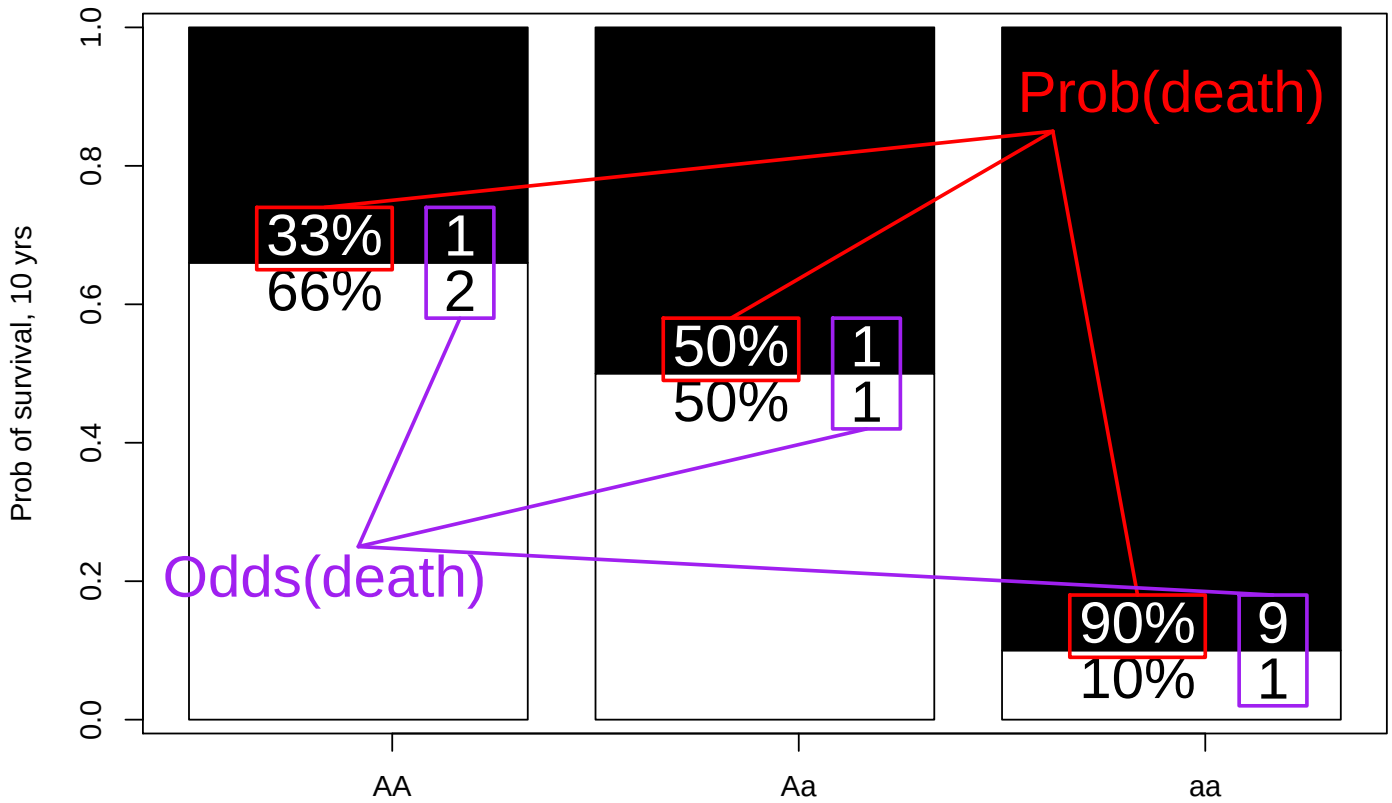
Interpreting odds

What is the odds of death for each group?



Interpreting odds

What is the *odds ratio* of death for group Aa relative to group AA?



Odds ratio for Aa vs AA: $\frac{\text{Odds(death} \mid \text{Group Aa)}}{\text{Odds(death} \mid \text{Group AA)}} = \frac{1/1}{1/2} = 2$

*The odds of death in group Aa is **two times** the odds of death in group AA*

Logistic regression: predicting malignancy

Logistic regression fits a linear model for the logarithm of the *odds* that $Y = 1$:

$$\log\left(\frac{Pr(Y_i = 1|X_i = x_i)}{Pr(Y_i = 0|X_i = x_i)}\right) \quad = \quad \beta_0 + \beta_1 X_i$$

Interpretation: *The odds that $Y = 1$ among those who have a given value of X is estimated to be e^{β_1} times the odds of $Y = 1$ among those who have a unit less of X*

This is not necessarily an intuitive way to think about modeling binary data. **Want more practice?** Consider taking a course that introduces generalized linear models.

Logistic Regression in R

The function `glm` with `family="binomial"` performs logistic regression

```
glm1 <- glm(y ~ Cell.size, family="binomial", data = bc_y)
```

Syntax:

- `glm(response ~ predictor, family="binomial", data = dataset)`
  - `response` is between 0 and 1 (inclusive)

Logistic Regression in R

The output of a `glm` is similar to the output of `lm`. We can print out the results as follows:

```
glm1
```

```
##
## Call:  glm(formula = y ~ Cell.size, family = "binomial", data = bc_y
##
## Coefficients:
## (Intercept)    Cell.size
##      -4.960        1.489
##
## Degrees of Freedom: 698 Total (i.e. Null);  697 Residual
## Null Deviance:       900.5
## Residual Deviance: 275.6     AIC: 279.6
```

Logistic Regression in R

We can get more information as follows:

```
glm1 %>% summary
```

```
##
## Call:
## glm(formula = y ~ Cell.size, family = "binomial", data = bc_y)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.1081  -0.2474  -0.2474   0.0099   2.6465
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -4.960      0.360  -13.78   <2e-16 ***
## Cell.size      1.489      0.121   12.30   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 900.53  on 698  degrees of freedom
```

Logistic Regression in R

The coefficients of the variables are *log* odds ratios. So, for a confidence interval for the log odds ratio of `Cell.size`:

```
confint(glm1)
```

```
## Waiting for profiling to be done...
```

```
##                   2.5 %    97.5 %
## (Intercept) -5.718134 -4.300687
## Cell.size    1.266087  1.741961
```

All the extractor functions we saw before for linear regression are available for logistic regression (`coef`, `fitted`, `predict`...)

Logistic Regression in R

To obtain an estimate of the odds ratio and the corresponding 95% confidence interval for the odds ratio, just exponentiate:

```
glm1$coefficients %>% exp
```

```
## (Intercept)   Cell.size
## 0.007011619 4.431374991
```

```
confint(glm1, "Cell.size") %>% exp %>% round(3)
```

```
## Waiting for profiling to be done...
```

```
##  2.5 % 97.5 %
##  3.547  5.709
```

The odds of malignant diagnosis are estimated to be 4.43 times higher for a one unit increase in Cell size.

Logisitic regression

We can fit a logistic regression with multiple covariates by adding the variables to the model with +:

```
glm(y ~ Cell.shape + Cell.size,
    family="binomial", data = bc_y)
```

```
##
## Call:  glm(formula = y ~ Cell.shape + Cell.size, family = "binomial"
##     data = bc_y)
##
## Coefficients:
## (Intercept)   Cell.shape    Cell.size
##     -5.4771       0.8223       0.7672
##
## Degrees of Freedom: 698 Total (i.e. Null);  696 Residual
## Null Deviance:       900.5
## Residual Deviance: 239.6     AIC: 245.6
```

Other regressions

Other `family` arguments provide other forms of regressions. Some of the regression analyses available with `glm` are:

- `family = poisson`
  - Often useful if you have count data as a response
- `family = Gamma(link = "log")`
  - Often useful if you have a response that is positively skewed
- `family = Gamma(link = "inverse")`
  - Often useful if you have a response that can only be positive

Remember last week's disclaimer – it is important to understand the models you are fitting; please take care (and perhaps courses) before using the above.

Summary

- Linear models can be fit with interactions between variables, and without intercepts
- Logistic regression can be performed with `glm()` with `family="binomial"`
  - Logistic regression fits a linear model on the log-odds scale
  - To obtain odds and odds ratios, you exponentiate the coefficients
- There are many other regression methods (generalized linear models) available using `glm()`, including Poisson and Gamma regression

Next week: advanced data manipulation, including working with multiple datasets

The plan

- 5 minute break
- In-class exercise available via Canvas
  - Designed to be completed by 3:20 p.m.
  - Due today 6:30 p.m.
  - *Yellow sticky note* = urgent; *blue sticky note* = non-urgent
- Homework due next week by 1 p.m. Friday
- Office hours as always: Tuesday, Wednesday and Thursday