FEEG6017 lecture: Logistic regression

Dr. Brendan Neville
bjn1c13@ecs.soton.ac.uk

Regression with a binary outcome variable

- Previous lecture: simple linear regression, with one continuous variable (height) being used to predict another (basketball ability).
- This lecture: logistic regression. We try to predict results on a binary outcome variable using one or more predictor variables.

Binary outcome variables

- This comes up often in science.
- We might want to predict whether:
  - patients will live or die
  - species will thrive or go extinct
  - structures will fail or stay standing
- ... based on their scores on a set of potential predictor variables.

Example data set

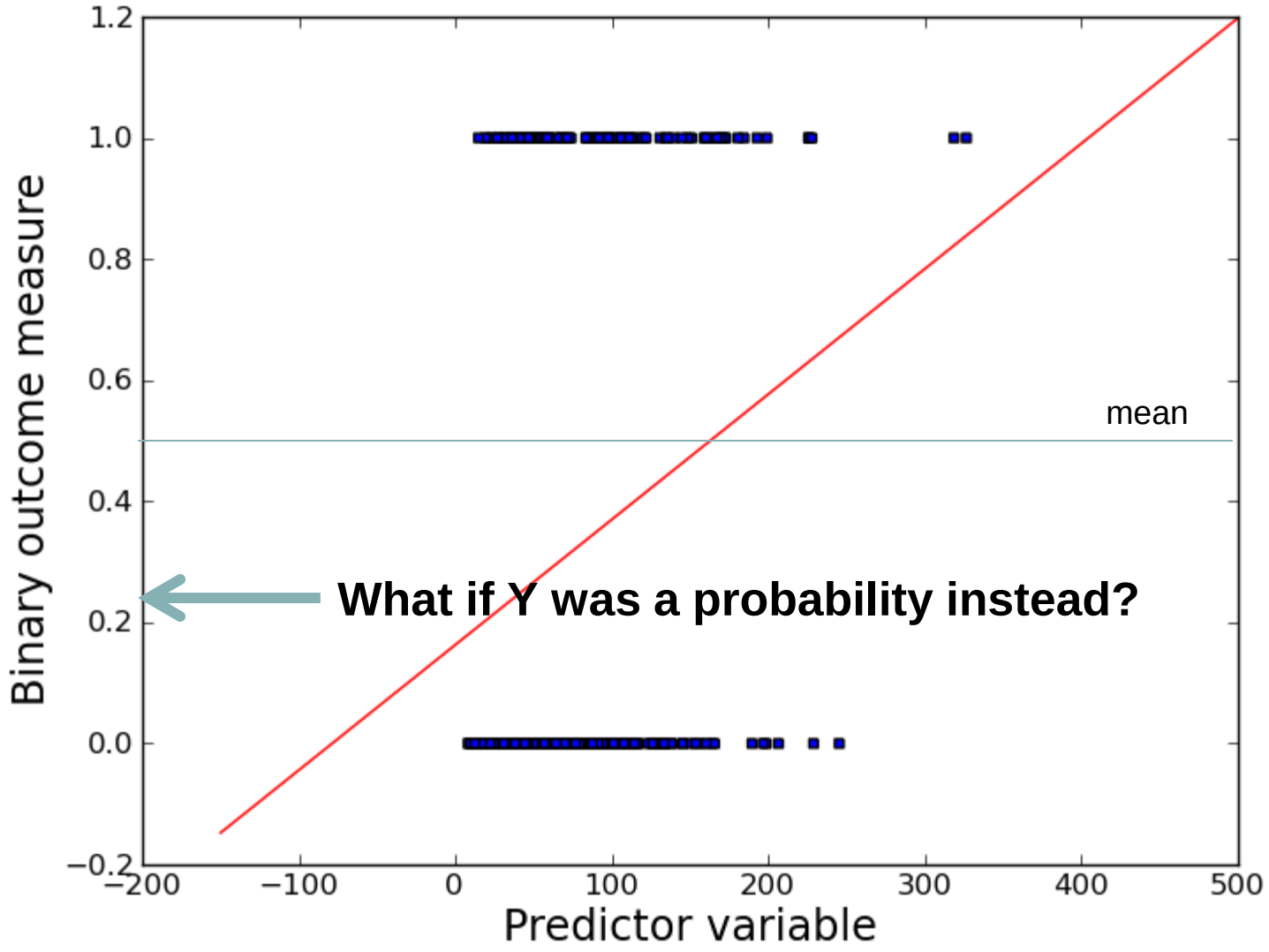
Fate	Treatment	Age(2005)
alive	placebo	45
dead	drug A	61
dead	placebo	29
alive	drug B	33
dead	placebo	70
dead	drug A	61
dead	placebo	44
alive	drug B	50

Dealing with a binary outcome

- Looking at the example data, we might try to predict patient survival based on which drug group they were in, much like an ANOVA.
- We might also try to predict survival based on the age of the patient at the start of the study, much like a regression.
- But there's a problem.

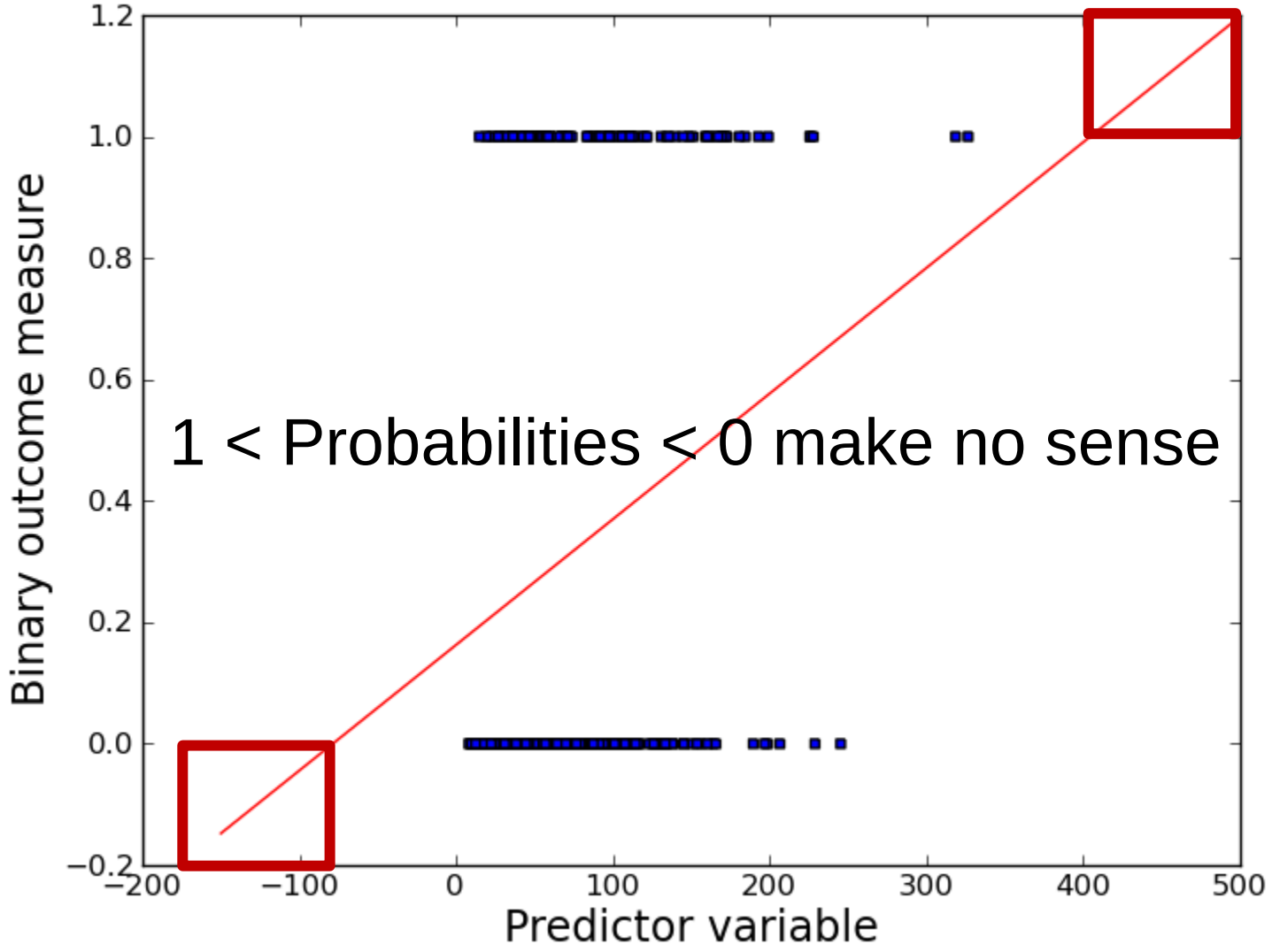
Dealing with a binary outcome

- “alive" or "dead" is a binary state.
- If we code "alive" as 1 and "dead" as 0, can we just fit a line through those points as we would with linear regression?
- It's possible. We could then try to interpret values on the fitted line as the probability of survival.


1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
Binary outcome measure
mean
What if Y was a probability instead?
−200
−100
0
100
200
300
400
500
Predictor variable

Dealing with a binary outcome

- The trouble is we're dealing with a probability and yet the regression can easily give us illegal values less than 0 or greater than 1.


1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
Binary outcome measure
−200
−100
0
100
200
300
400
500
Predictor variable
1 < Probabilities < 0 make no sense

Dealing with a binary outcome

- Assume:
  - P(X) >1 implies P(X) = 1
  - P(X) < 0 implies P(X) = 0
- A danger of "saturation": once we predict a patient has a 100% chance of survival, we're maxed out and no new fact about the patient could improve their odds.

Can we fix this?

- The trick is that we can transform the outcome variable to something which we can use linear regression on.
- This would have a valid range of values in the range –infinity to + infinity.
- But still be related to the “probability”.

Odds

- Say I have a bag of 100 marbles
- 20 are blue, 10 are red 10 are green, 60 yellow.
- 20% probability of picking a blue one
- 20 chances of picking a blue one
- 80 chances of pick another colour
- So the odds of picking a blue marble are 20/80 = 2/8 = 0.25

From probability to odds

- Another way of thinking about probabilities is to transform them using the function:

$$\text{Odds} = p / (1 - p) = 0.2/0.8 = 0.25$$

- This is the probability of something happening divided by the probability of it not happening

From probability to odds

- Odds are commonly used in gambling, especially horse-racing.
  - "9 to 1 against", meaning a probability of 0.1.
  - "Even odds", meaning p = 0.5.
  - "3 to 1 on" meaning p = 0.75.
  - **“8 to 2 against” meaning odds = 0.25**
  - **“4 to 1 against” meaning odds = 0.25**
  - Don’t write up your work like this.

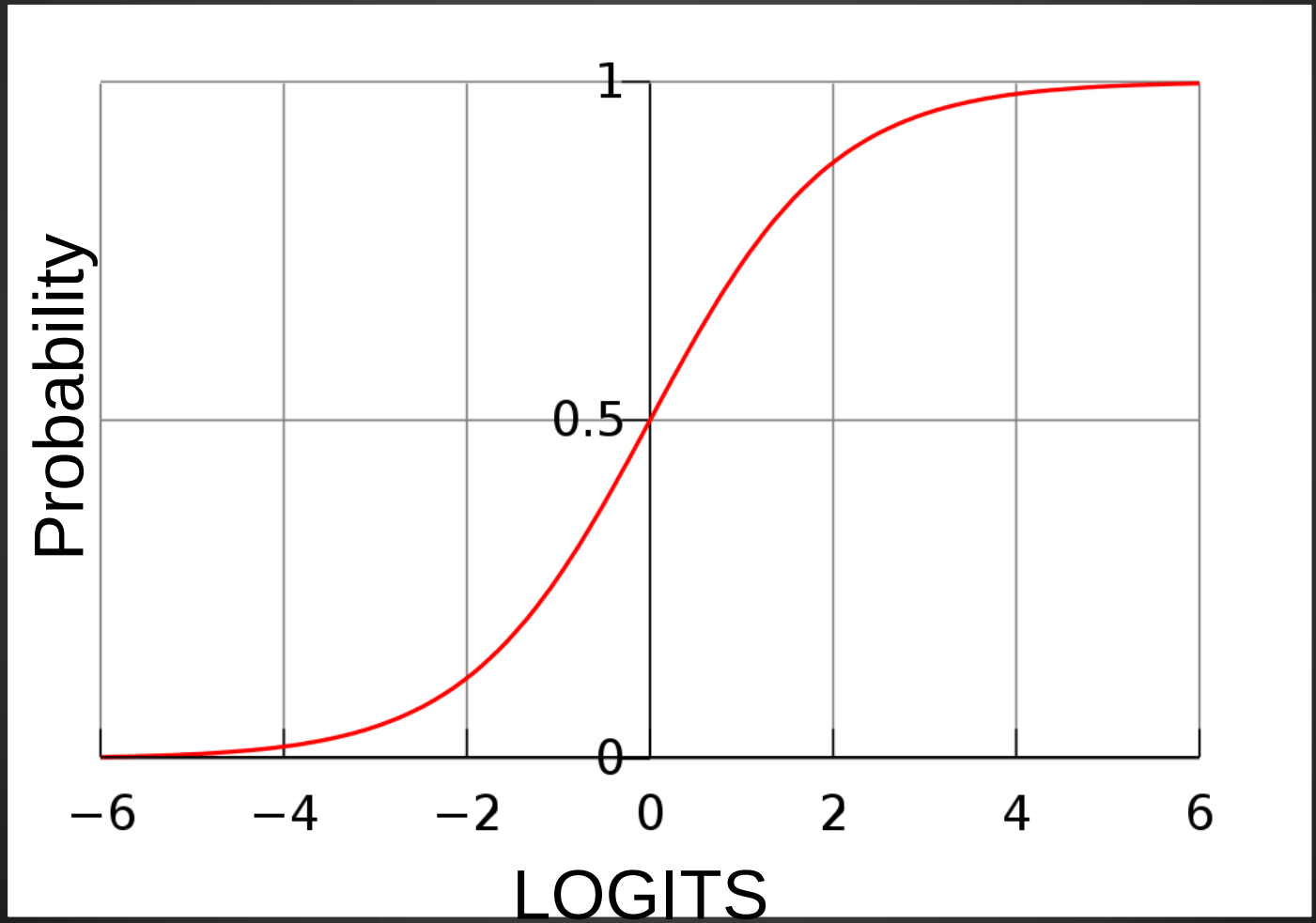
Odds values for some probabilities

Probability	Odds
1.0	Undefined
0.99	99
0.75	3
0.5	1
0.25	0.3333
0.01	0.0101
0.0	0.0

The logistic function

- Odds give us a value that ranges from 0 (for very small probabilities) to positive infinity (for probabilities close to 1.0).
- With one more transformation we can get a value that is unbounded over the real numbers.
- This is the logistic function: $y = \log(p / (1-p))$.
- y is measured in logits.

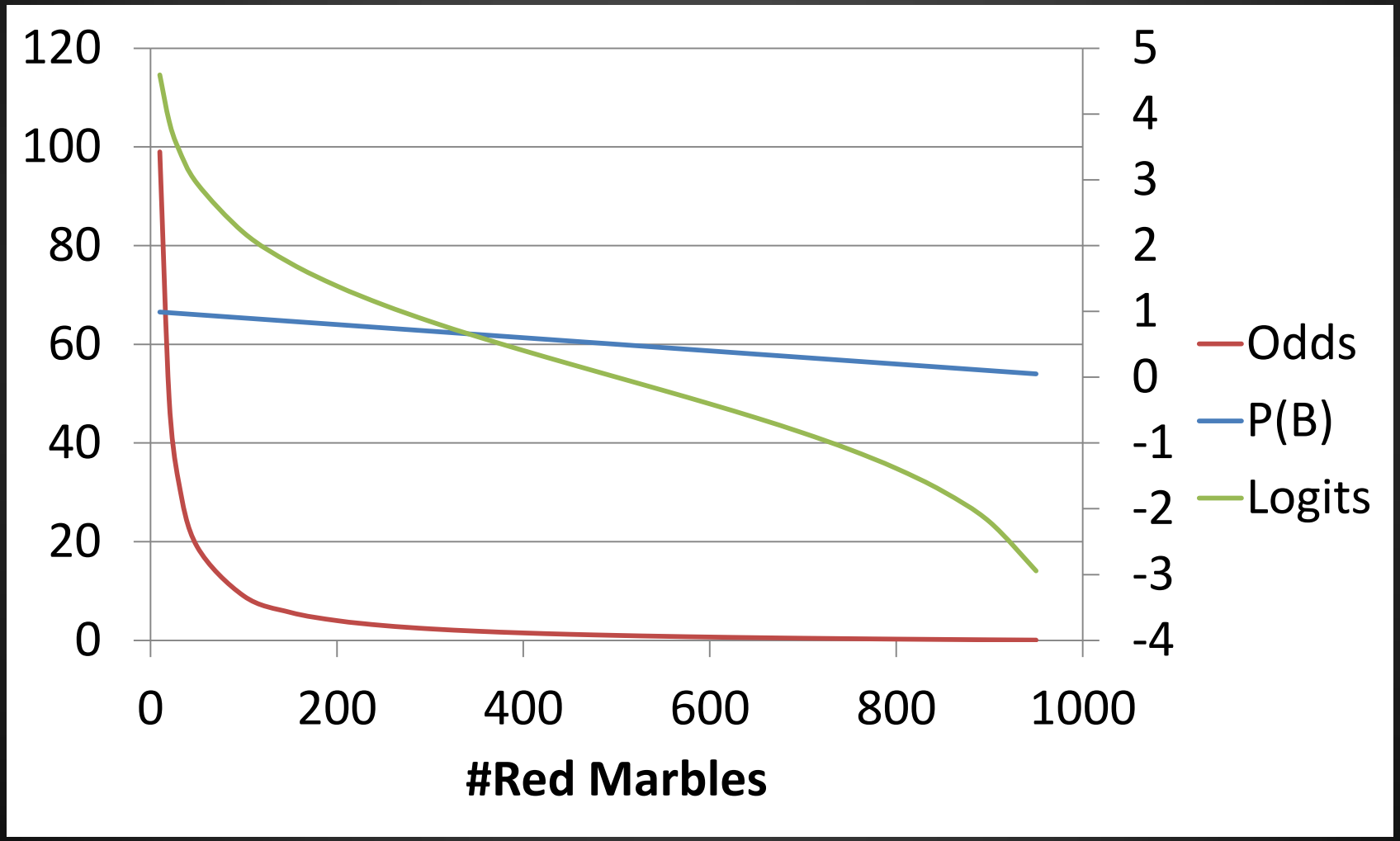
From Logits to Probability



Different measures of Outcome

- Probability of the outcome
  - Range [0,1]
  - No use linear regression needs [–inf, +inf]
- Odds
  - Realtive probability of an event happening vs. the probability of it not happening
  - $O_{alive}=P_{alive}/1-P_{alive} = P_{alive}/P_{dead}$
  - Range [0,+infinity]
- Logit values
  - $Logit_{alive} = ln(O_{alive})$
  - Range [-infinity, +infinity]

Odds, Probs and Logits



Logistic regression

- Probabilities transformed through the logistic function are known as "logit" values.
- We now have a value that ranges from negative infinity to positive infinity: ideal for linear regression.

Logit values for some probabilities

Probability	Logit score
0.99	1.996
0.75	0.477
0.5	0.0
0.25	-0.477
0.01	-1.996

Relationship to linear regression

- A regression can be done on the inferred logit values just as for any other continuous variable.

- However: by having to use the logistic function as a "linking function" between what our data says and the underlying model, we've moved from simple linear models to generalized linear models.

Implied model of data generation

Intercept

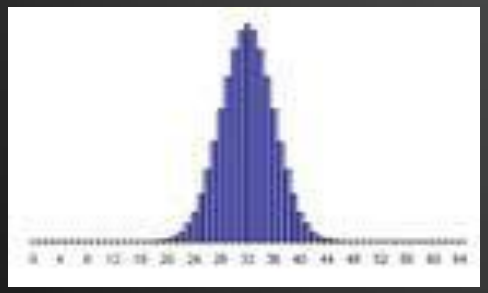
+

Slope

X

Predictor variable

+



Random noise

=

Logit score

→

Probability of event

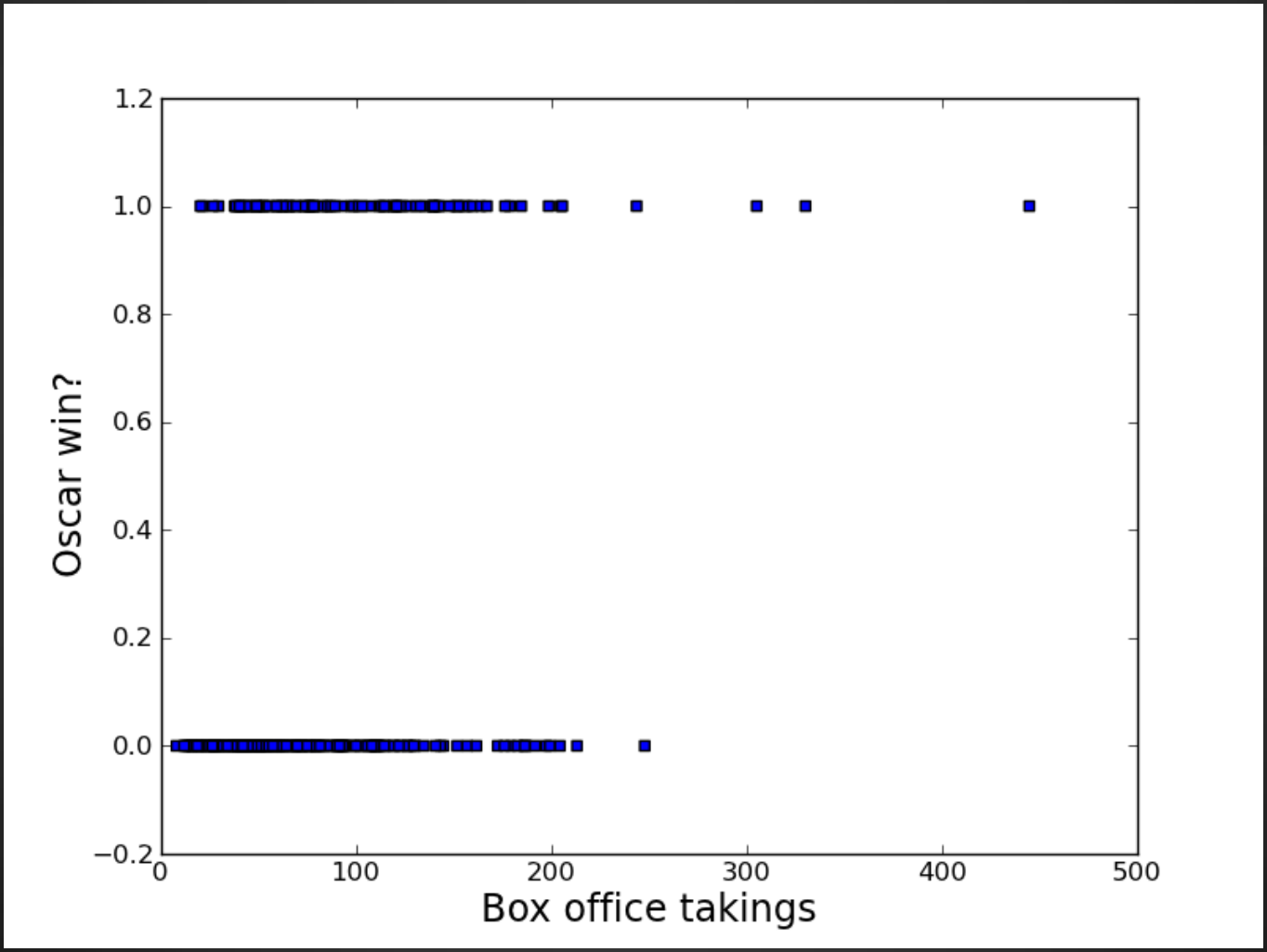
Introducing the Oscars example

- A fictional data set that looks at the question of what it takes for a film to win an Oscar.
- Outcome variable: Oscar win, yes or no?
- Predictor variables:
  - Box office takings in millions of dollars.
  - Budget in millions of dollars.
  - Country of origin: US, UK, Europe, India, other.
  - Critical reception (average score 0-100).
  - Length of the film in minutes.

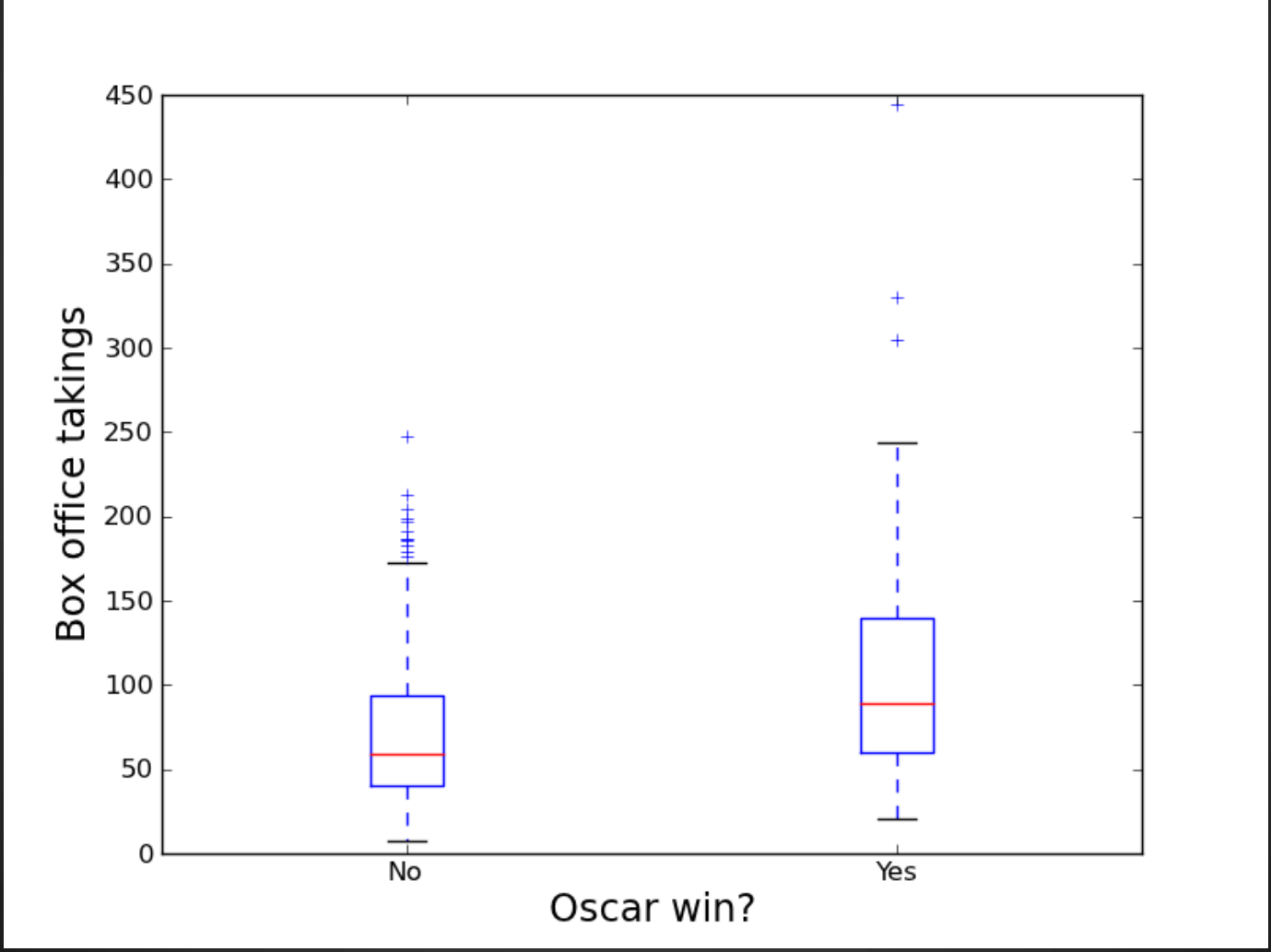
Predicting Oscar success

- Let's start simply, by looking at only one of the five predictor variables.
- **So: do big box office takings make Oscar success more likely?**
- Same Technique to use:
- Budget Size
- Film length

Oscar success by box office takings



Oscar success by box office takings



The logic of logistic regression

- Data suggests that big box office takings may increase the chance of winning an Oscar.
- We will therefore try to predict the probability of an Oscar win as a function of the film's box office takings.
- We can't work with raw probabilities in a regression so we will switch to logit scores.

How do I run a logistic regression in R?

- Read the data into R in the usual way; key variables are `Oscar` and `BoxOffice`.
- We build the logistic regression model with:

```
boxOfficeModel = glm( Oscar ~ BoxOffice, family=binomial (link="logit"))
```

- GLM stands for "generalized linear model".
- `summary(boxOfficeModel)` to see results.

Interpreting the R output

```
Call:
glm(formula = Oscar ~ BoxOffice, family = binomial(link = "logit"))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6432  -0.8316  -0.6997   1.2380   1.8546

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.750349   0.256883  -6.814 9.50e-12 ***
BoxOffice    0.011306   0.002507   4.510 6.48e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Interpreting the R output

- Output is similar to linear regression: difference is that we predict logit scores, and not the outcome measure directly.
- The intercept coefficient is -1.75. This gives us the logit score for a film that made $0.
- The BoxOffice coefficient is 0.011. This tells us the increase in logit score per $million in box office takings.

Interpreting the R output

- A z-test indicates that the p-value for the BoxOffice coefficient is 6.48×10^{-6}.
- This indicates that the data would be unlikely to arise in a world where the null hypothesis was true, i.e., in which there was no relationship between box office takings and Oscar success.

From logit score to probability

- So we can say that the logit score for a film that took $50 million is:

  -1.75 + (50 x 0.011) = -1.2

- But how to get from a logit score of -1.2 to a probability?
- Ultimately that's what we're trying to predict: the chances of a film that takes $50 million at the box office going on to win an Oscar.

From logit score to probability

- We need to work our way back through the logit score's construction process.
- exp(-1.2) = 0.301. This is an odds value.
- prob = odds / odds + 1.
- p = 0.301 / 1.301 = 0.231.
- Thus the model says our film has a 23.1% chance of winning an Oscar.

More than one predictor variable

- Remember there were five predictor variables: box office takings, budget, country of origin, critical reception, and length.
- *Could* look at each variable with a separate logistic regression to see whether or not it helps to explain Oscar success.
- A better approach is to fit all the variables in a single model.

More than one predictor variable

- The R command for fitting a multi-predictor model is easy:

```
fullModel = glm( Oscar ~ BoxOffice + Budget + Country + Critics + Length, family=binomial (link="logit"))
```

- This is not unique to logistic regression; we can do the same with simple linear regression.

More than one predictor variable

- By fitting all the variables together, we can look at how much of the variance in the outcome measure that they *jointly* explain.
- But possibly we won't want to use *all* the variables we have at hand. Typically some are predictively useful but others are not.
- This brings us to the topic of model reduction, for next week.

Additional materials

- The Python code for generating the fictional Oscars data set.
- The fictional Oscars data set as a text file.
- An R script for analyzing the fictional data set.