Week 12 Hour 2 – Logistic Regression examples

Example:

Consider a logistic model of cancer rates as a function of age.

```
glm(formula = cancer ~ age, family = binomial)
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -5.98270    1.88954  -3.166  0.00154 **
age          0.07378    0.02955   2.497  0.01252 *
```

The estimates and standard errors are given in terms of log odds, not in terms of probability.

The p-values mean the same thing they always have.

```
glm(formula = cancer ~ age, family = binomial)
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -5.98270    1.88954  -3.166  0.00154 **
age          0.07378    0.02955   2.497  0.01252 *
```

This means three things:

1. The coefficients can't be interpreted as simply as 'the chance of cancer at age 0'. We have to use the ***inverse logit*** in order to find that.

Log-odds of -5.98 is the same as probability ***0.00252***.

```
> ilogit(-5.98)
[1] 0.002522447
```

```
glm(formula = cancer ~ age, family = binomial)
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -5.98270    1.88954  -3.166  0.00154 **
age          0.07378    0.02955   2.497  0.01252 *
```

2. It is possible to get an intercept of less than 0.

This means that the ***log-odds*** are negative, NOT the probability.

A log-odds of 0 translates to a probability of ***0.5***.

```
glm(formula = cancer ~ age, family = binomial)
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -5.98270    1.88954  -3.166  0.00154 **
age          0.07378    0.02955   2.497  0.01252 *
```

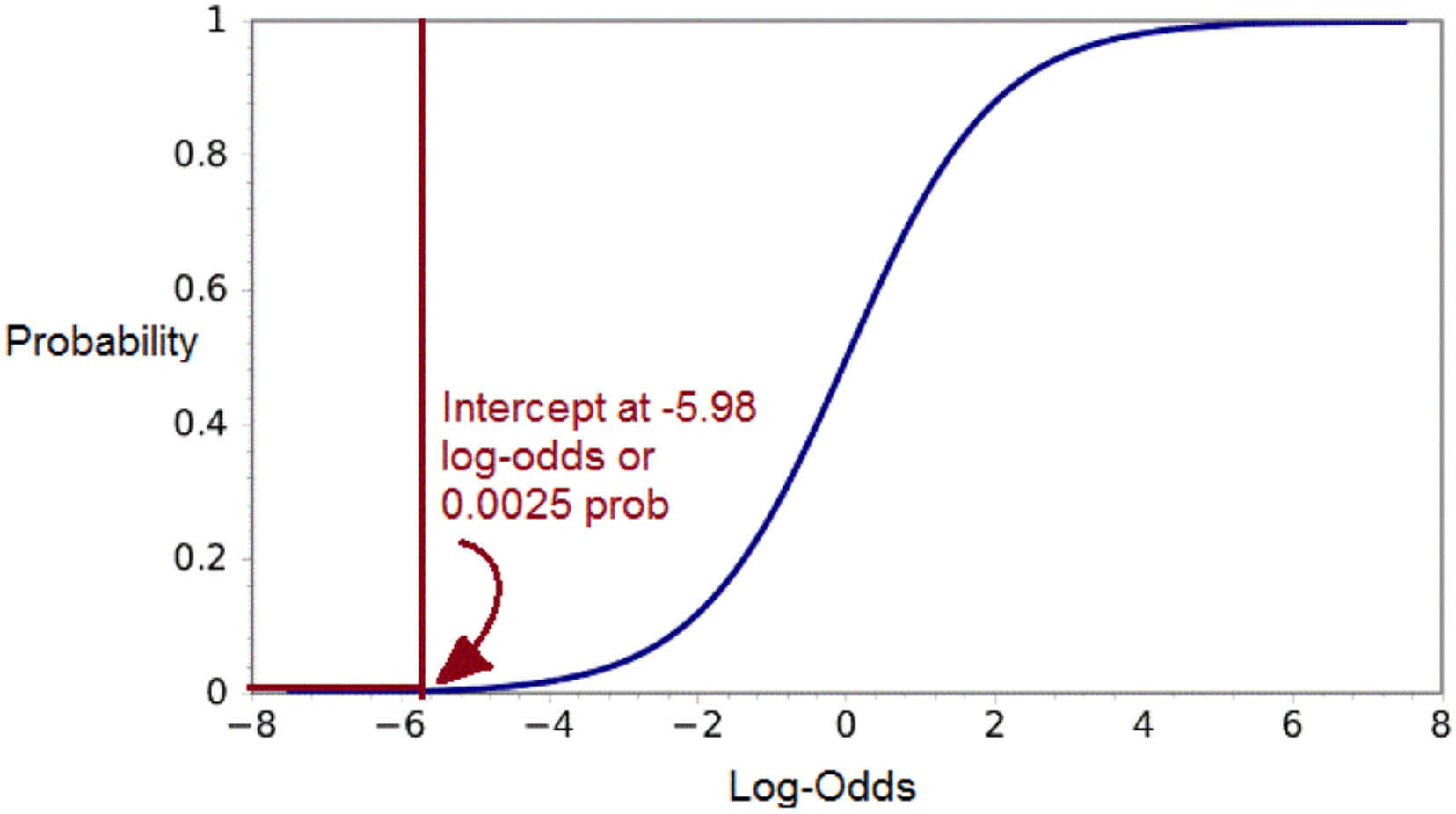
3. Log-odds always increases as probability increases.

Therefore…

A positive slope coefficient means that the response increases with the associated explanatory variable.

In this case, the probability of cancer increases with age.

… but it increases by the sigmoid curve, not at a constant rate.



Logistic models have regression formulas. This model's formula is:

Log-Odds(Cancer) = - 5.98 + 0.074(Age) + error

We can plug age values into this formula to get predicted log-odds at different ages.

Log-odds at age 45

$$-5.98 + 0.074*(45) = \mathbf{-2.65}$$

Log-odds at age 60

$$-5.98 + 0.074*(60) = \mathbf{-1.54}$$

Log-odds isn't particular useful to interpret, but we can use the inverse logit function to turn log-odds back into probability.

Cancer probability at age 45

Inv. Logit(-2.65) = ***0.0659***

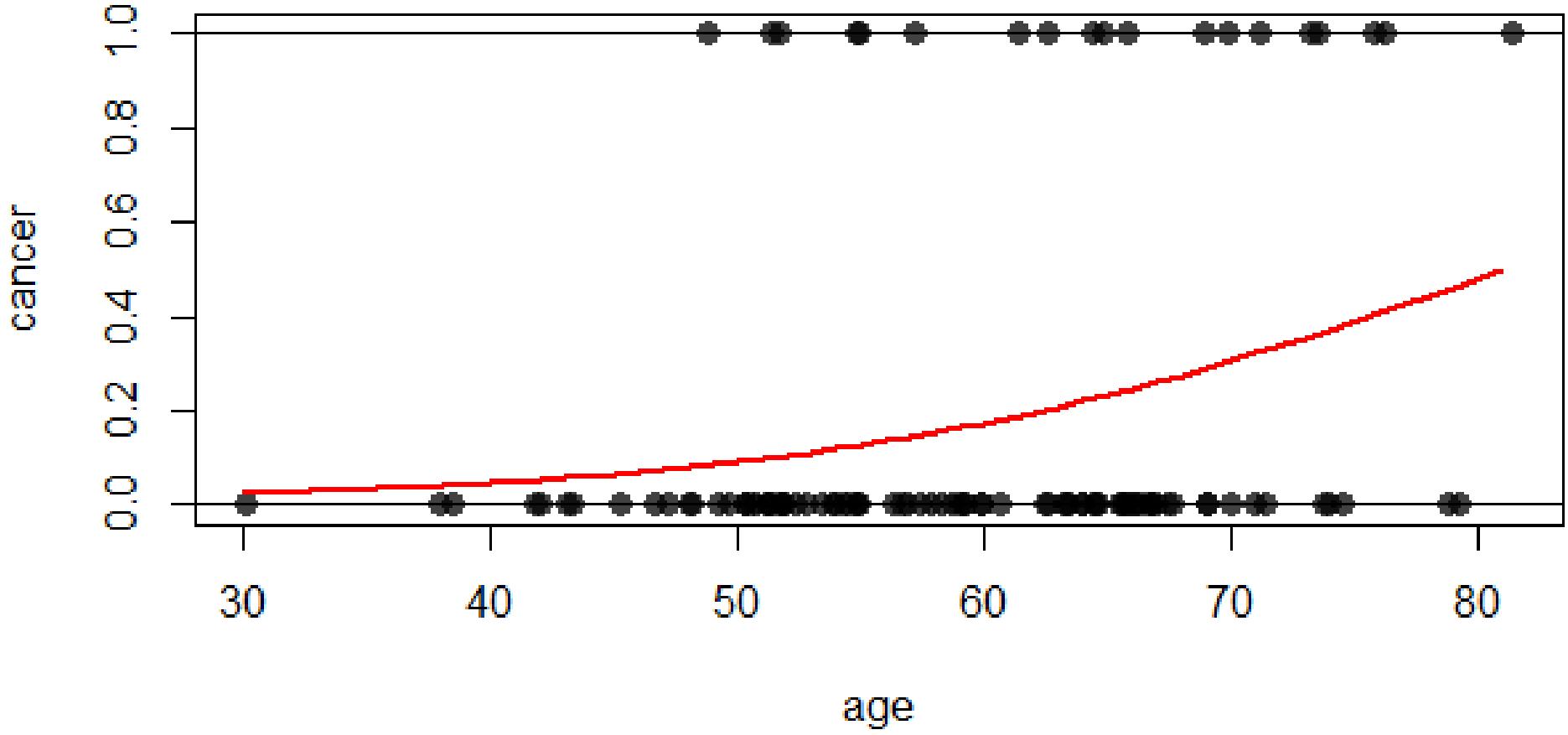
Cancer probability at age 60

Inv. Logit (-1.54) = ***0.1765***

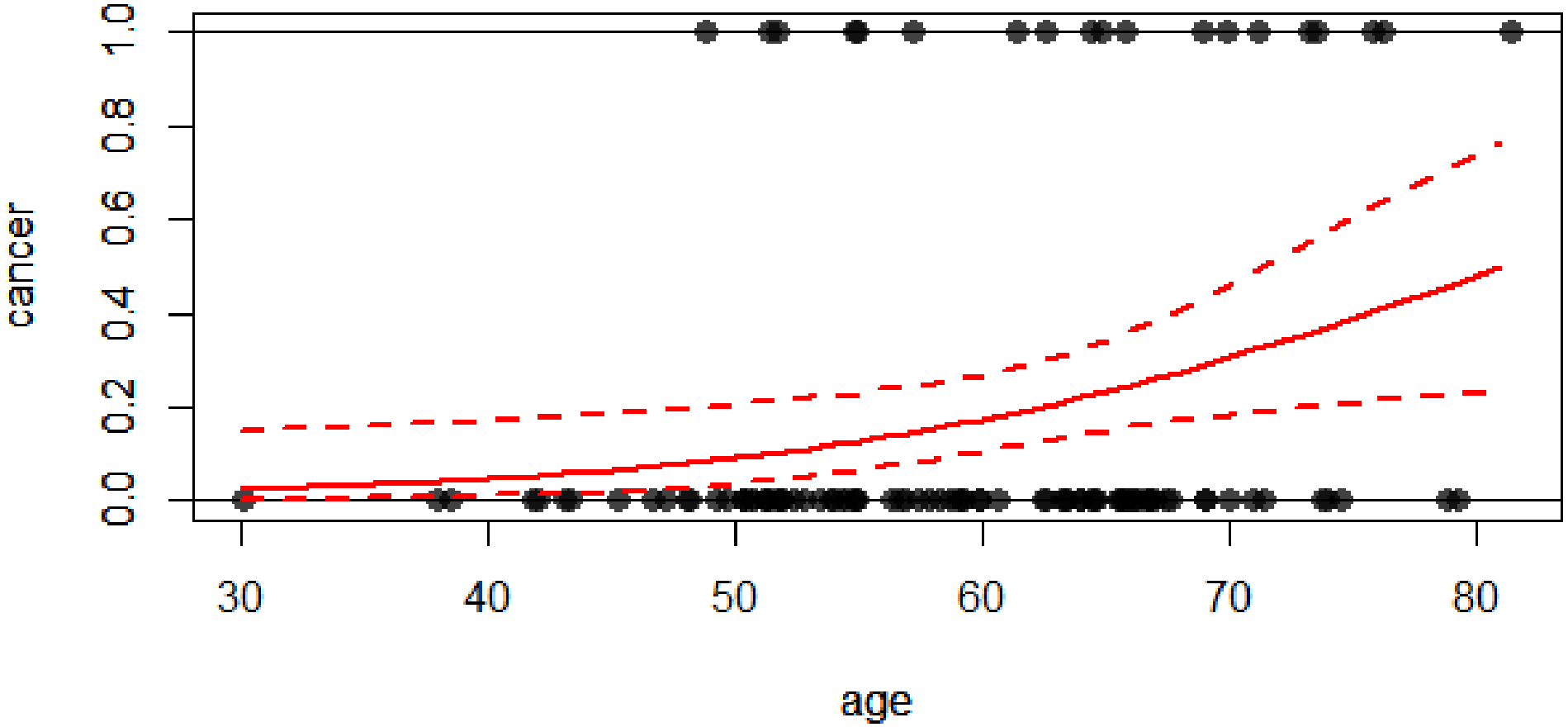
Cancer probability at age 75

Inv. Logit (-5.98 + 0.074*(60)) = Inv. Logit (-0.43) = ***0.3941***

Every fitted value stays within the [0,1], and has a reasonable interpretation!



The confidence intervals stay within as [0,1] as well, and widen as we move away from one of the certainties.



One good tern deserves another.

Let’s try another example.

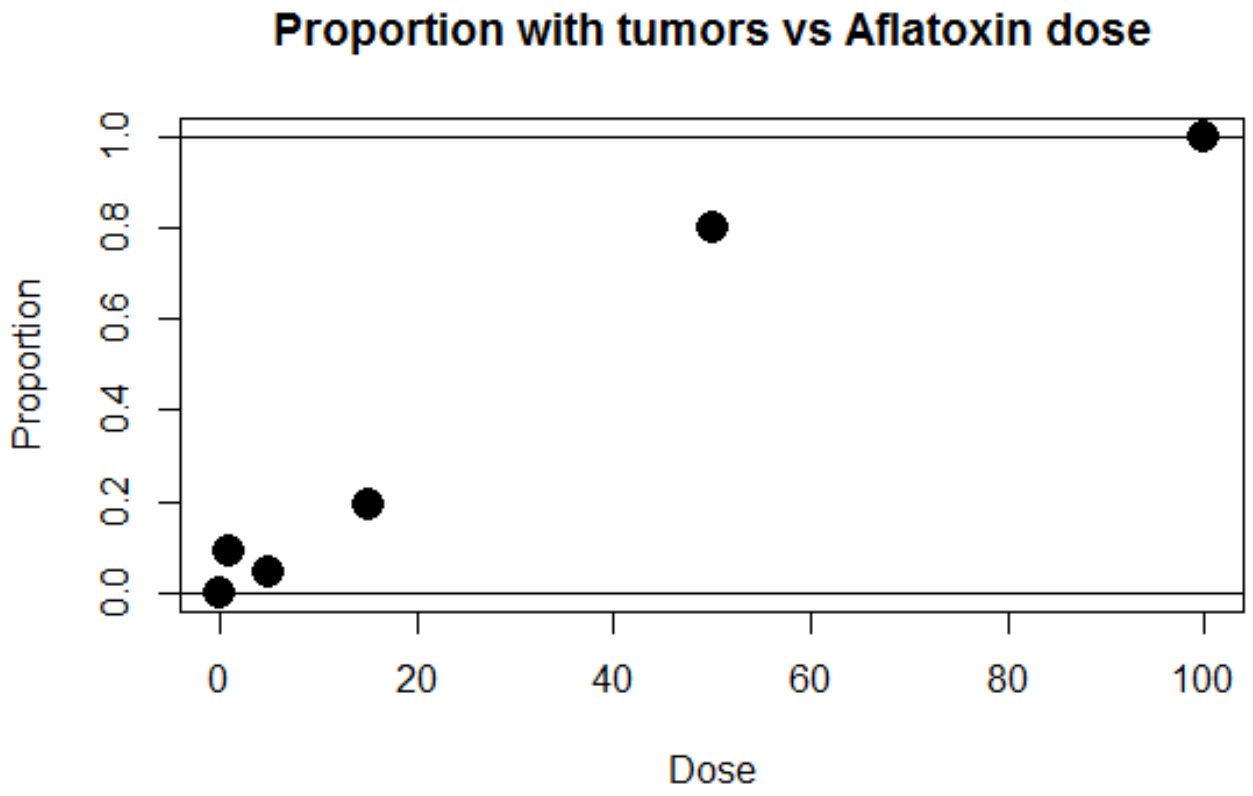
Consider the dataset aflatoxin in the faraway package.

```
> aflatoxin
  dose total tumor
1    0    18     0
2    1    22     2
3    5    22     1
4   15    21     4
5   50    25    20
6  100    28    28
```

In the dataset, 0/18 animals that received 0 dose of the toxin developed liver tumors.

2/22 animals that received 1 dose developed tumors, etc.

The proportion that developed tumors increases with the dosage. Our goal is to develop a model that fits a probability of a tumor at any dosage level.



```
glm(formula = cbind(tumor, no_tumor) ~ dose, family = binomial,
    data = afla)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.03604    0.48226  -6.295 3.07e-10 ***
dose         0.09009    0.01456   6.189 6.04e-10 ***
```

Building a logistic model, we find…

Intercept: The log-odds of a tumor at dose=0 are -3.036

Dose: The log-odds of a tumor increase by 0.090 for every unit of dosage increase.

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.03604    0.48226  -6.295 3.07e-10 ***
dose         0.09009    0.01456   6.189 6.04e-10 ***
```

So our logistic regression equation is

Log odds(Tumor) = -3.036 + 0.090(Dose) + error

Which we can use to see how a dose of 7 would do..

Log odds(Tumor|Dose = 7) = -3.036 + 0.090(7)

= -2.405

Pr(Tumor | Dose = 30) = Inv Logit(-2.405) = 0.0828

Or a dose of 30…

$$\text{Log odds}(\text{Tumor}|\text{Dose} = 30) = -3.036 + 0.090(30)$$

$$= -0.3333$$

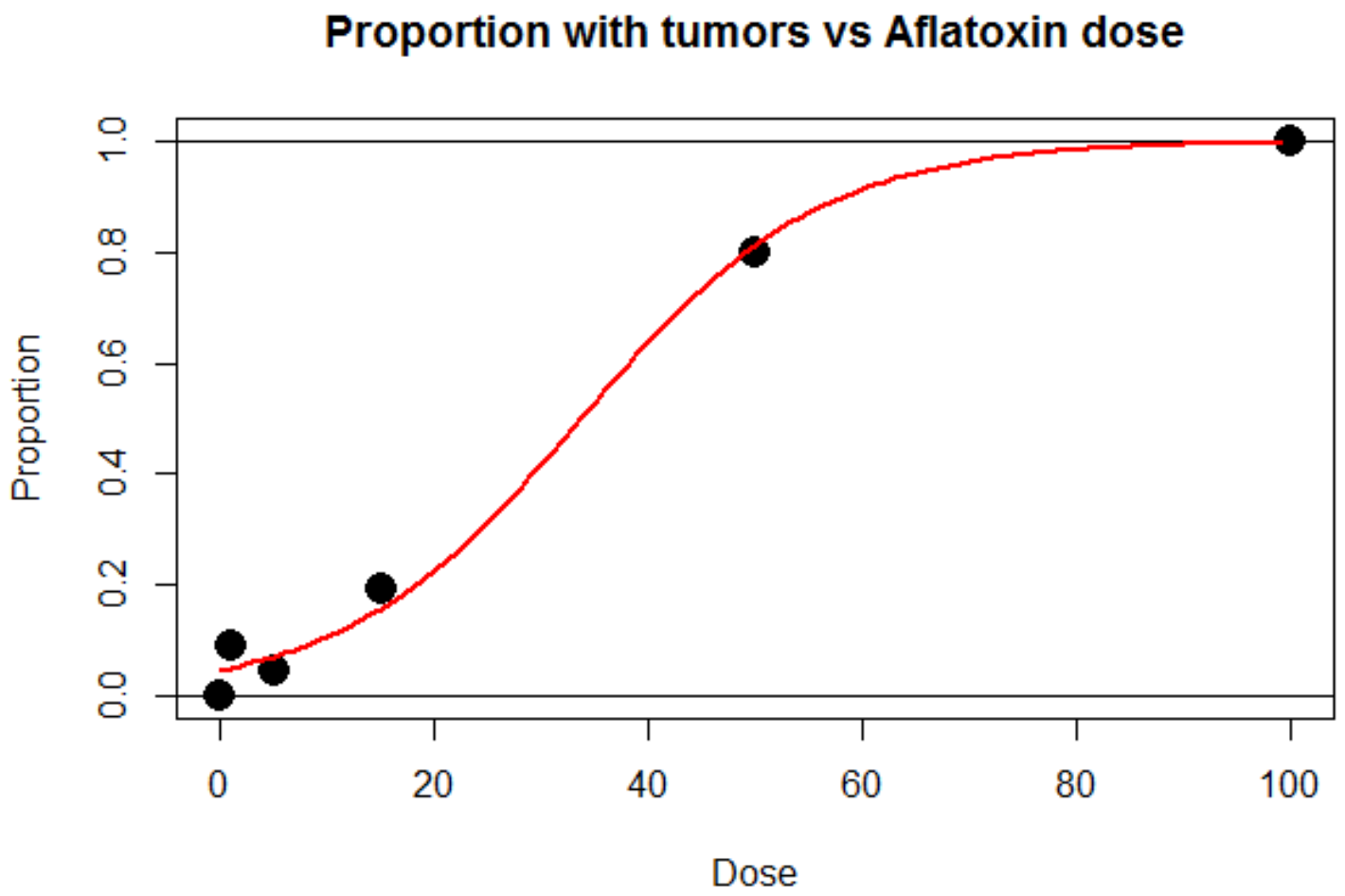
$$\Pr(\text{Tumor} \mid \text{Dose} = 30) = \text{Inv Logit}(-0.3333) = 0.4174$$

Or a dose of 200… (beyond the original data)

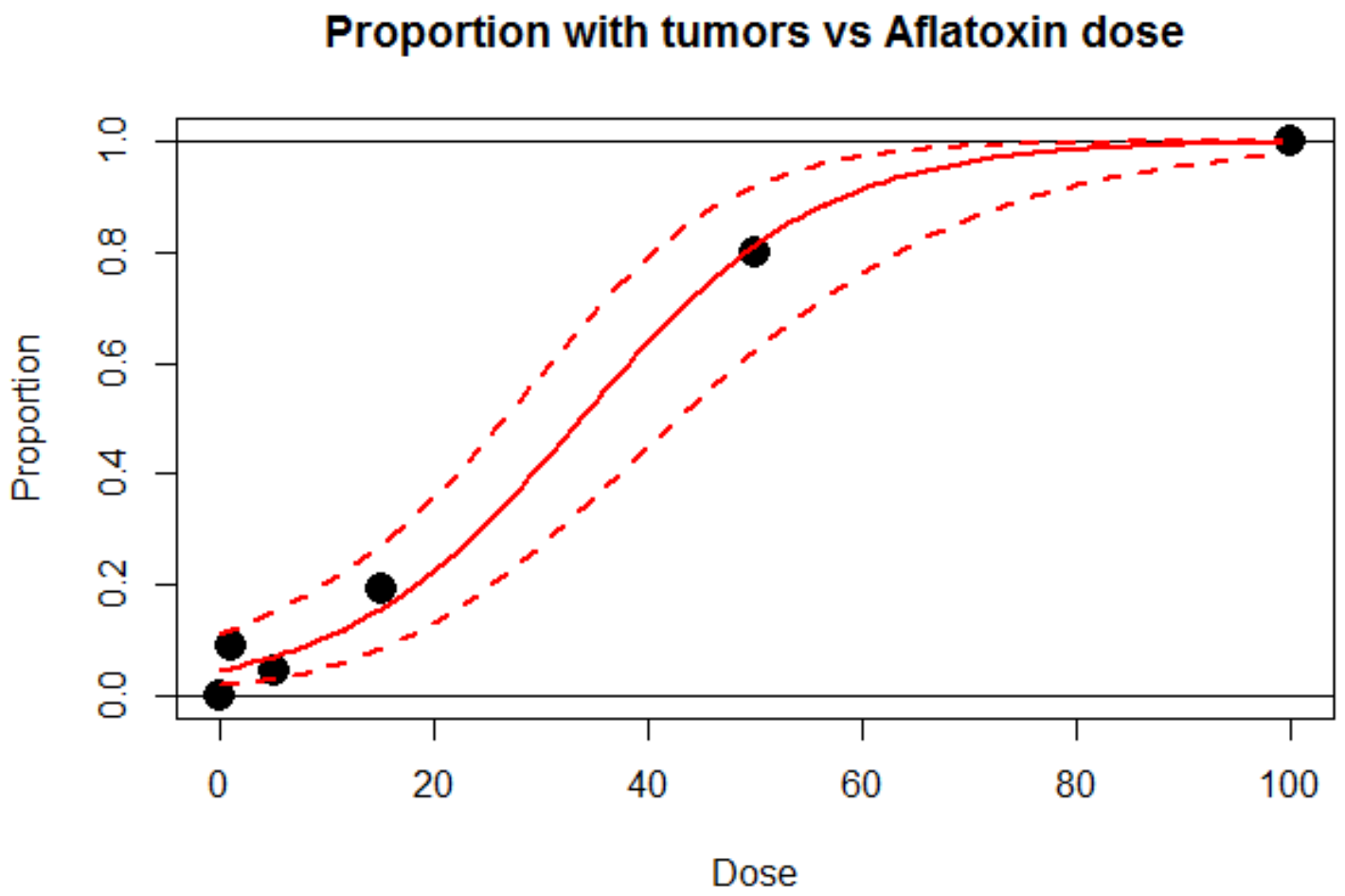
$$\text{Log odds}(\text{Tumor}|\text{Dose} = 200) = -3.036 + 0.090(200)$$

$$= 14.88$$

$$\Pr(\text{Tumor} \mid \text{Dose} = 30) = \text{Inv Logit}(14.88) = \mathbf{\mathit{0.9999}}$$



The sigmoid curve is more apparent here than in the last example.



At both extremes, the upper and lower confidence intervals remain well-behaved.

How do we get such good results from only six data points?

We didn't.

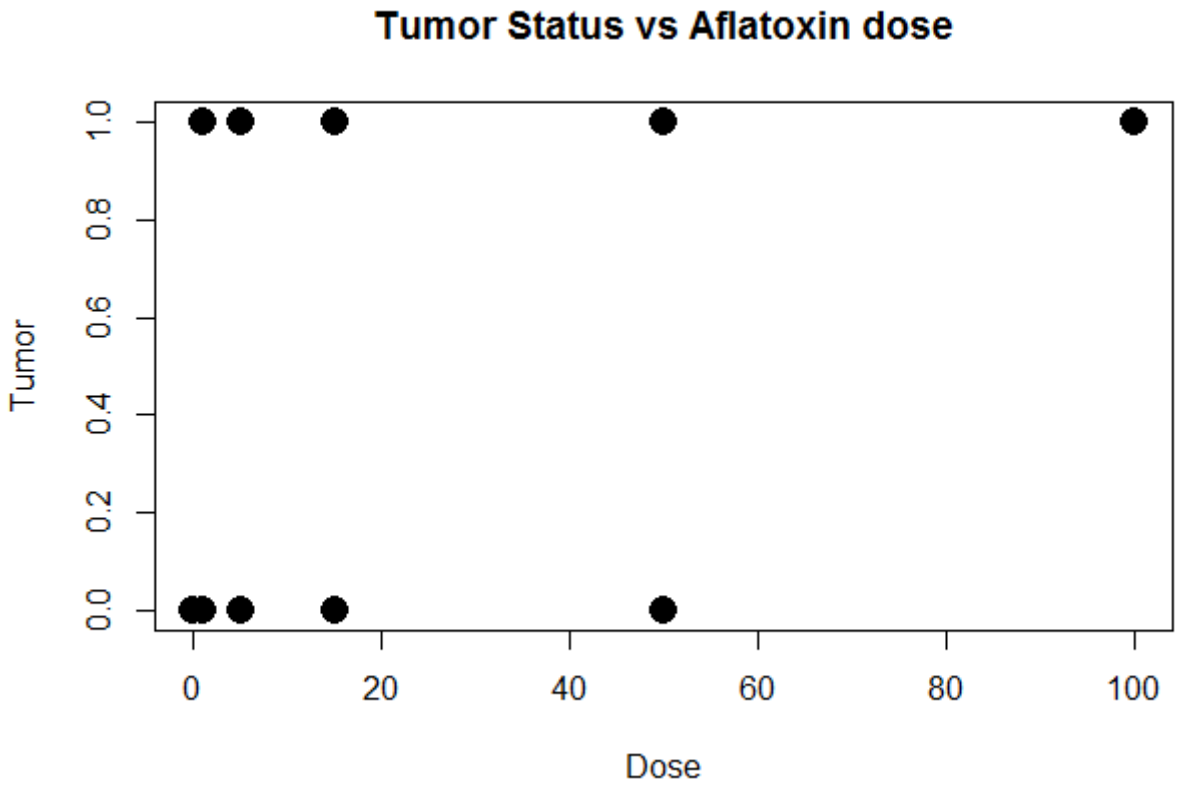
The sample size isn't 6. It just happens that the response has been replicated several times at each dosage level.

n= 18 + 22 + 22 + 21 + 25 + 28 = 136. One for each animal.

```
> aflatoxin
  dose total tumor
1    0    18     0
2    1    22     2
3    5    22     1
4   15    21     4
5   50    25    20
6  100    28    28
```

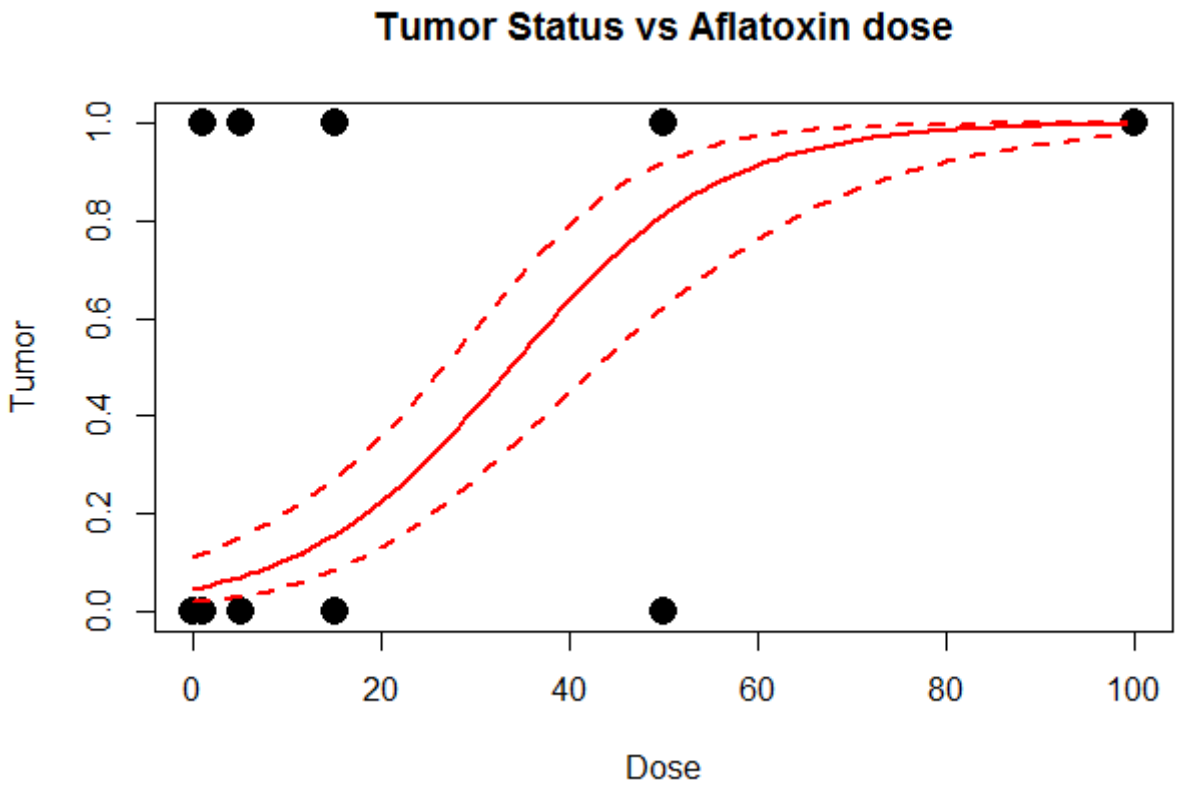
A dataset with 136 rows with the dose and a 0 or 1 response for the tumor variable would have worked the same way.

Plotting these values on the tumor / dose axes looks like this:



Because of the overlapping points, it's hard to tell that tumors increase with dosage.

A model from such data, and the curves, would be the same



Logistic Regression can handle either format for data. The case-by-case data that you're used to (left),

Or the data that has been grouped into identical cases (right).

The identical cases is more common in laboratory settings.

	age	cancer
1	56	0
2	47	0
3	74	0
4	69	1
5	55	0
6	57	1

```
> aflatoxin
  dose total tumor
1    0    18     0
2    1    22     2
3    5    22     1
4   15    21     4
5   50    25    20
6  100    28    28
```

Manatees: Nature experts on gentle curves

Comparing Logistic Regression to Linear Regression

The most important similarity:

Logistic regression uses the same explanatory variable structure as linear regression.

Logistic regression can use transformed variables, interactions, and dummy variables, as well as more than one explanatory variable at a time. Just like linear regression.

The most important difference:

Everything is in terms of log-odds, not of probability.

Consider the intercept in the aflatoxin example. We can find a quick 95% confidence interval of the intercept, the tumor chance at dose 0, with the estimate and standard error.

Estimate +/- 1.96*Std. error. =

-3.036 +/- 1.96* 0 .4823 =

-3.981 to -2.091

To convert this confidence interval into a probability, we can take the inverse logit of the lower and upper bounds.

But all the real work, such as finding critical values and the adding/subtracting the margin of error, happens with log odds.

Inv. Logit(-3.981) = ***0.0183***

Inv. Logit(-2.091) = ***0.1100***

Hypothesis tests work the usual way in logistic regression.

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.03604    0.48226  -6.295 3.07e-10 ***
dose         0.09009    0.01456   6.189 6.04e-10 ***
```

In the aflatoxin example,

For both coefficients, $p < 0.001$.

We have very strong evidence that the intercept is non-zero.

We have very strong evidence that dos is non-zero.

Also, since dose is positive, we have very strong evidence that the chance of a tumor increases as dose increases.

Like other models, logistic regression has an AIC (and a BIC).

This means that model select methods can be used to find the 'best' set of explanatory variables in a logistic regression.

The mechanics surrounding the logit transform make it impossible to compare the AIC from a logistic to the AIC from a linear and find the best model

There are modifications to the R-squared for logistic, including Naglekirke's R-squared and McFadden's R-squared.

To use linear regression in R, use the lm() command and specify the model and the data being used.

```
 mod_linear = lm( cbind(tumor,no_tumor) ~ dose
, data=afla)
```

To use logistic regression in R, use the ***glm()*** command, and specify the model and data AND ALSO use the ***family=binomial*** setting.

```
mod_logistic = glm( cbind(tumor,no_tumor) ~ dose,
data=afla, family=binomial)
```

Part of logistic regression is automatically assign weights to individual values. That means not every observation is treated equally.

This makes concepts like degrees of freedom, and methods dependent upon df such as ANOVA, much messier.

We won't be addressing these issues further in class other than to show some R output right now:

In the summary data

Deviance is a loose replacement for sum of squares.

Null deviance is, very loosely, the amount of variation in the responses.

Residual deviance is the amount of variation left unexplained by the model.

```
    Null deviance: 116.524  on 5  degrees of freedom
Residual deviance:   2.897  on 4  degrees of freedom
```

…and in the ANOVA data

We see the same deviance. Here it's broken down so we can see the among of deviance explained by each term.

Notice that the degrees of freedom is listed as '5' in total, which is only a reflection of the fact that the observations were grouped into six rows. It is NOT a reflection of there being 136 independent observations.

	Df	Deviance	Resid. Df	Resid. Dev
NULL			5	116.524
dose	1	113.63	4	2.897

The predict() function can be used on logistic models to find the fitted values, and to predict responses to new sets of explanatory variables.

Like linear regression, the standard error of each estimate is found, but in terms of log odds.

	Fit	StdError
1	-3.036036	0.4822595
2	-2.945948	0.4714049
3	-2.585593	0.4301784
4	-1.684705	0.3489868
5	1.468402	0.4838024
6	5.972841	1.1379301