Classification

CS-309

Utility

- A person arrives at the emergency room with a set of symptoms that could possibly be attributed to one of three medical conditions.

- An online banking service must be able to determine whether or not a transaction being performed on the site is fraudulent, on the basis of the user's IP address, past transaction history, and so forth.

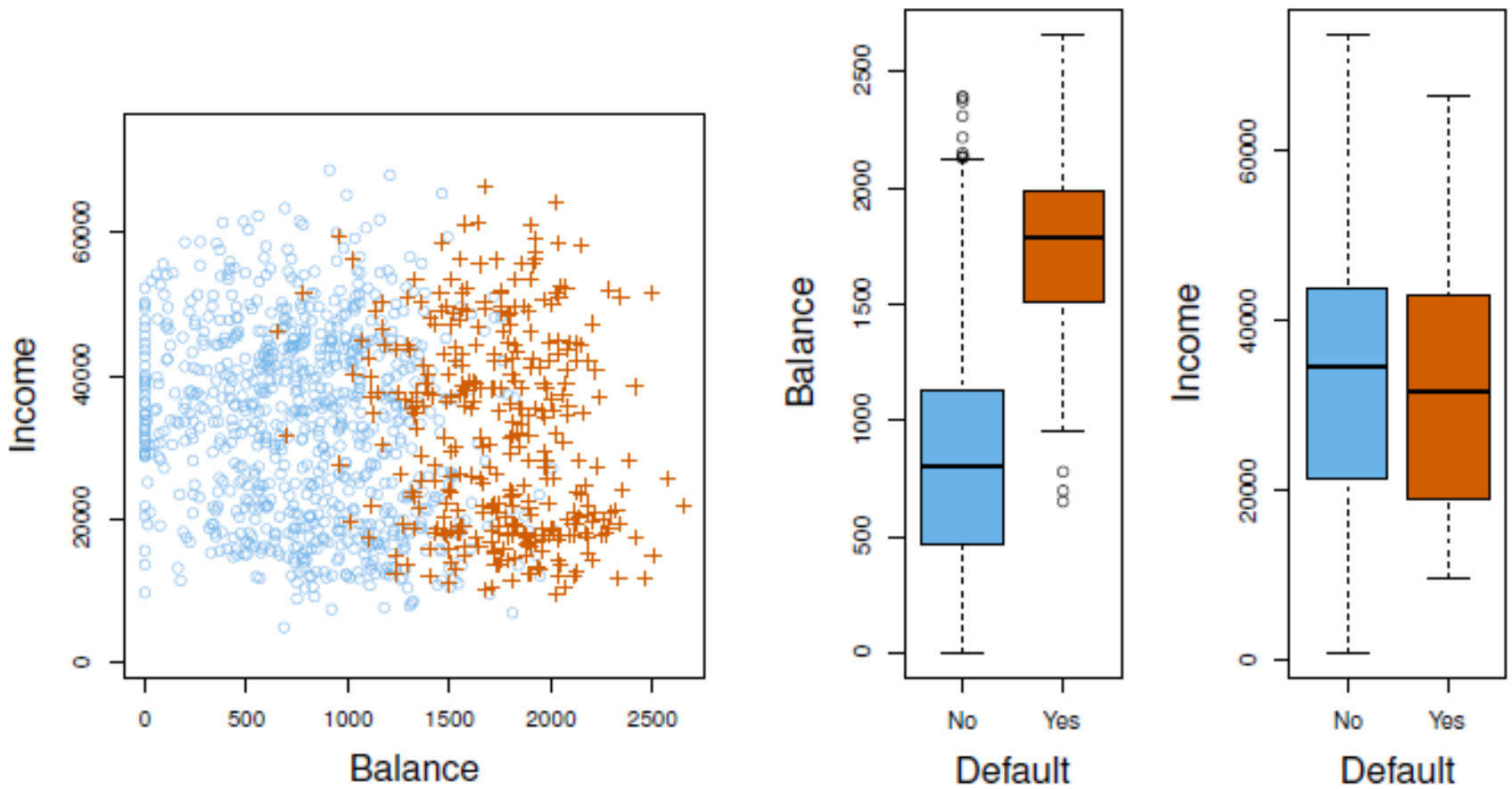
- On the basis of DNA sequence data for a number of patients with and without a given disease, a biologist would like to figure out which DNA mutations are deleterious (disease-causing) and which are not.

Classification

- Qualitative variables take values in an unordered set $\mathcal{C}$, such as:
  `eye color` $\in$ {`brown`, `blue`, `green`}
  `email` $\in$ {`spam`, `ham`}.
- Given a feature vector X and a qualitative response Y taking values in the set $\mathcal{C}$, the classification task is to build a function $C(X)$ that takes as input the feature vector X and predicts its value for Y; i.e. $C(X) \in \mathcal{C}$.
- Often we are more interested in estimating the *probabilities* that X belongs to each category in $\mathcal{C}$.

  For example, it is more valuable to have an estimate of the probability that an insurance claim is fraudulent, than a classification fraudulent or not.

Example: Credit Card Default



Why Not Linear Regression?

- Suppose that we are trying to predict the medical condition of a patient in the emergency room on the basis of her symptoms.
- In this simplified example, there are three possible diagnoses: stroke, drug overdose, and epileptic seizure.
- We could consider encoding these values as a quantitative response variable, Y , as follows:

$$Y = \begin{cases} 1 & \text{if } \texttt{stroke}; \\ 2 & \text{if } \texttt{drug overdose}; \\ 3 & \text{if } \texttt{epileptic seizure}. \end{cases}$$

- Unfortunately, this coding implies an ordering on the outcomes, putting drug overdose in between stroke and epileptic seizure, and insisting that the difference between stroke and drug overdose is the same as the difference between drug overdose and epileptic seizure.

- In practice there is no particular reason that this needs to be the case. For instance, one could choose an equally reasonable coding,

$$Y = \begin{cases} 1 & \text{if } \texttt{epileptic seizure}; \\ 2 & \text{if } \texttt{stroke}; \\ 3 & \text{if } \texttt{drug overdose}. \end{cases}$$

- If the response variable's values did take on a natural ordering, such as *mild*, *moderate*, and *severe*, and we felt the gap between mild and moderate was similar to the gap between moderate and severe, then a 1, 2, 3 coding would be reasonable.

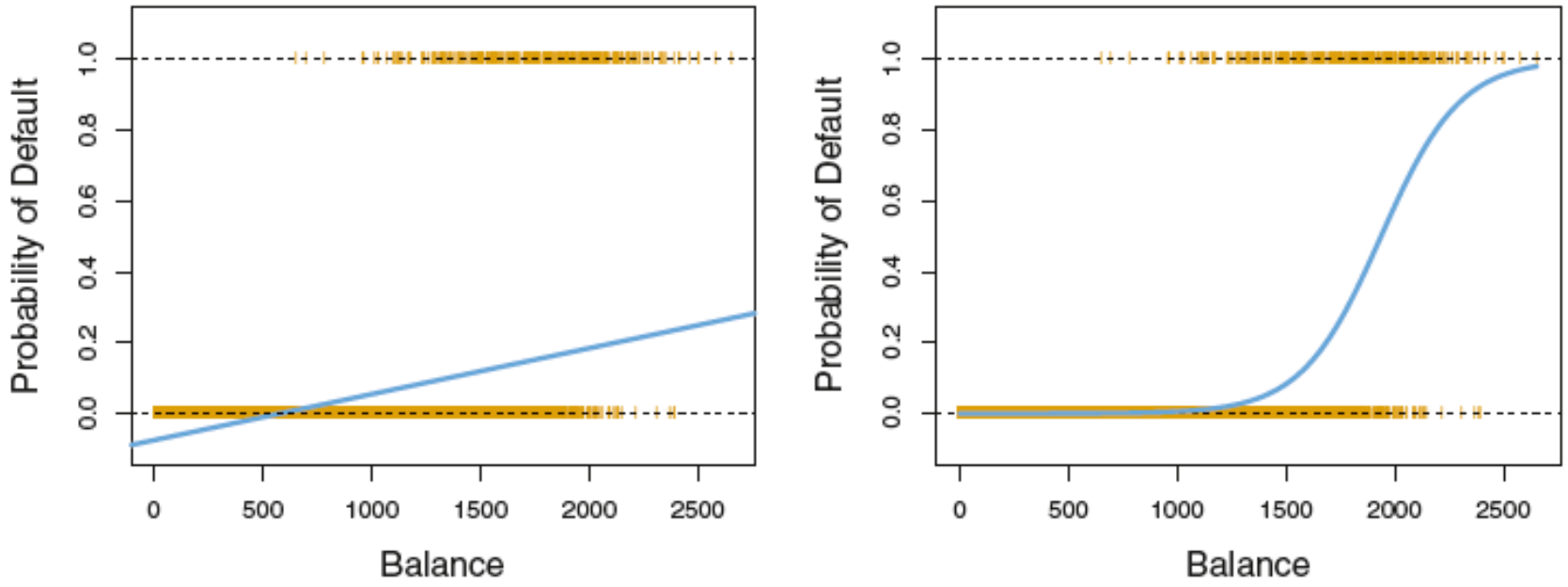
- Unfortunately, in general there is no natural way to convert a qualitative response variable with more than two levels into a quantitative response that is ready for linear regression.

- For a *binary* (two level) qualitative response, the situation is better. For instance, perhaps there are only two possibilities for the patient's medical condition: stroke and drug overdose.
- We could then potentially use the *dummy variable* approach to code the response as follows:

$$Y = \begin{cases} 0 & \text{if } \texttt{stroke}; \\ 1 & \text{if } \texttt{drug overdose}. \end{cases}$$

- For a binary response with a 0/1 coding as above, regression by least squares does make sense;
- it can be shown that the $X\hat{\beta}$ obtained using linear regression is in fact an estimate of Pr(drug overdose|*X*) in this special case.

However, if we use linear regression, some of our estimates might be outside the [0, 1] interval (see Figure), making them hard to interpret as probabilities.



Left: *Estimated probability of* default *using linear regression. Some estimated probabilities are negative! The orange ticks indicate the 0/1 values coded for* default*(*No *or* Yes*).* Right: *Predicted probabilities of* default *using logistic regression. All probabilities lie between* 0 *and* 1*.*

- Any time a straight line is fit to a binary response that is coded as 0 or 1, in principle we can always predict $p(X) < 0$ for some values of X and $p(X) > 1$ for others (unless the range of X is limited).
- To avoid this problem, we must model $p(X)$ using a function that gives outputs between 0 and 1 for all values of X.
- In logistic regression, we use the *logistic function*

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}. \qquad \frac{p(X)}{1-p(X)} = e^{\beta_0+\beta_1 X}.$$

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X$$

Estimating the Regression Coefficients

- The coefficients β_0 and β_1 are unknown, and must be estimated based on the available training data.
- *maximum likelihood* is preferred to estimate β_0 and β_1 .
- The basic intuition behind using maximum likelihood to fit a logistic regression model is as follows:
  - we seek estimates for β_0 and β_1 such that the predicted probability $\hat{p}(x_i)$ of default for each individual, corresponds as closely as possible to the individual's observed default status.
  - In other words, we try to find β_0 and β_1 such that plugging these estimates into the model for $p(X)$, yields a number close to one for all individuals who defaulted, and a number close to zero for all individuals who did not.

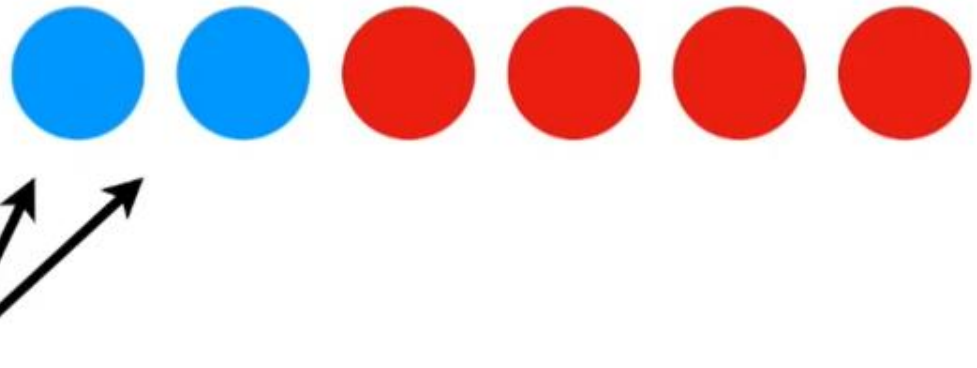
- This intuition can be formalized using a mathematical equation called a *likelihood function*

$$\ell(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} (1 - p(x_{i'}))$$

- The estimates for β_0 and β_1 are chosen to *maximize* this likelihood function.

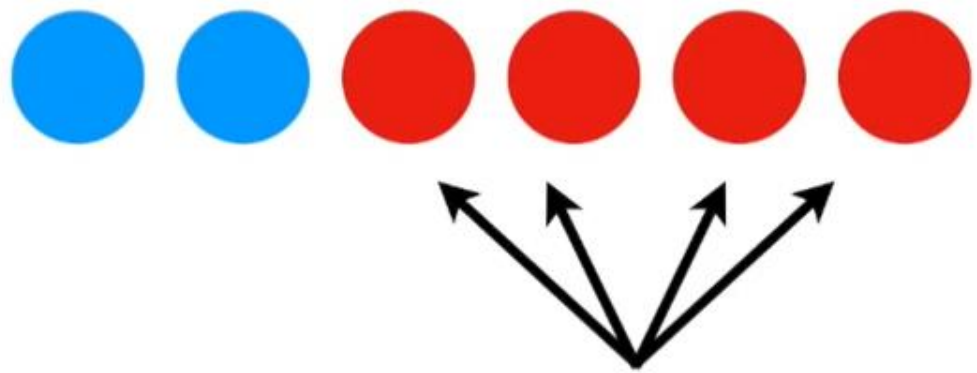
Odds and Log Odds

We illustrated this with circles…



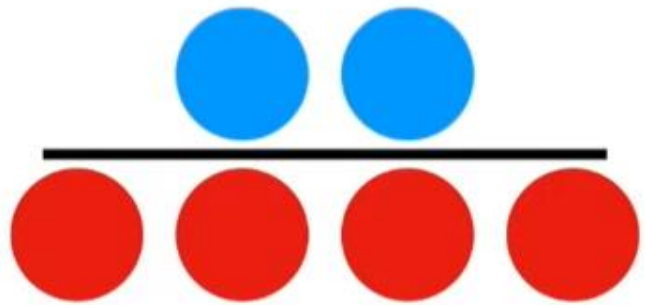
Blue circles represented my team **winning**.

We illustrated this with circles…

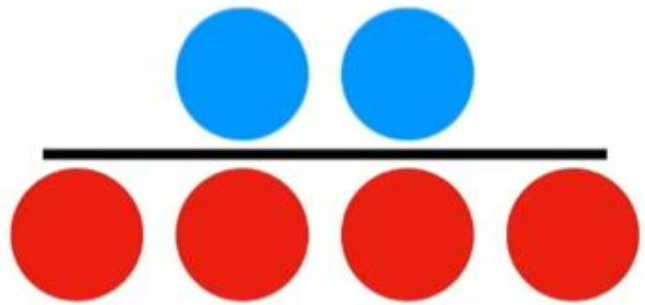


Red circles represented my team **losing**.

And the odds of my team **winning** were just the **blue** circles over the **red** circles…

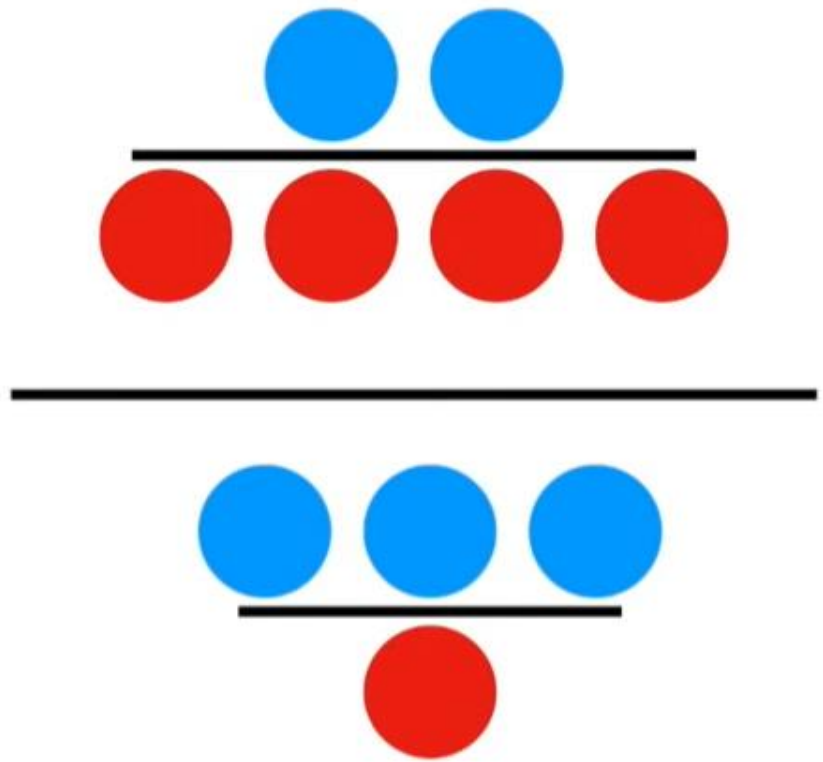


…and when we do the math we see the odds of **winning** are 0.5.



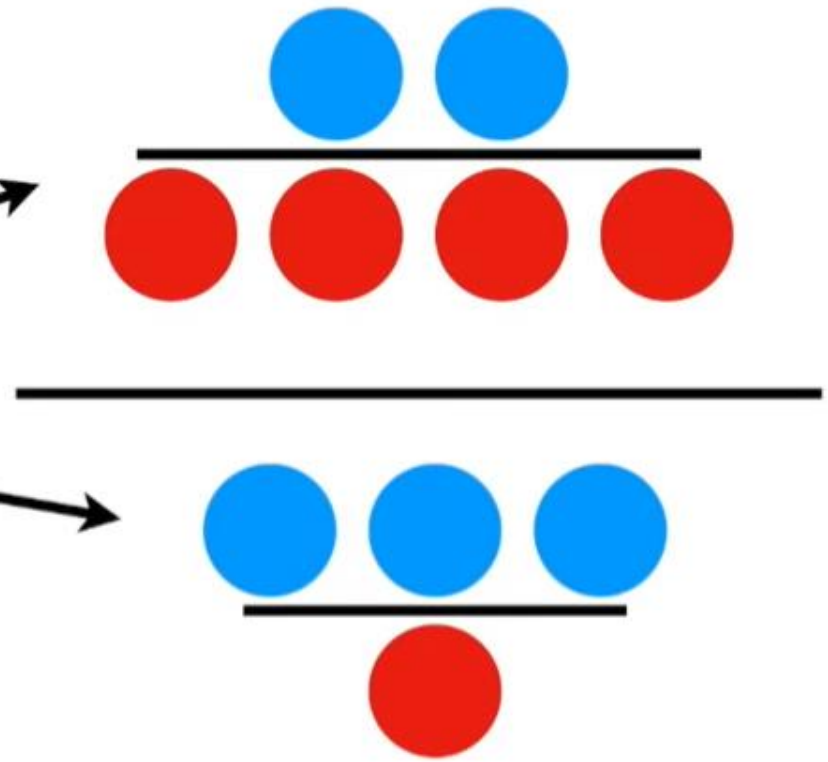
$$= \frac{2}{4} = 0.5$$

When people say “odds ratio”, they are talking about a “**ratio of odds**”.

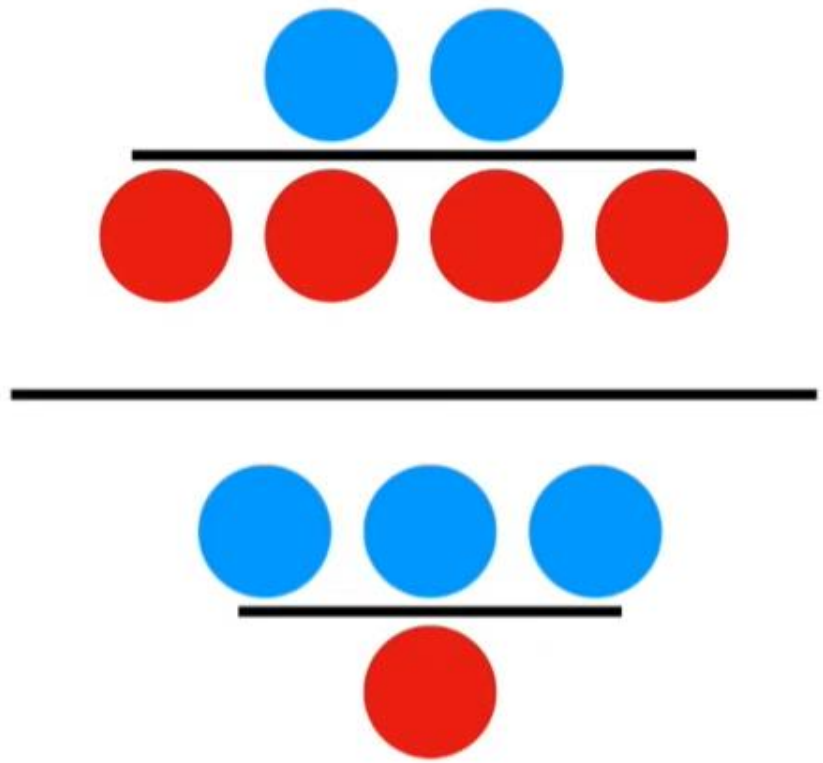


When people say “odds ratio”, they are talking about a “**ratio of odds**”.

So we’ve got a ratio of these odds…

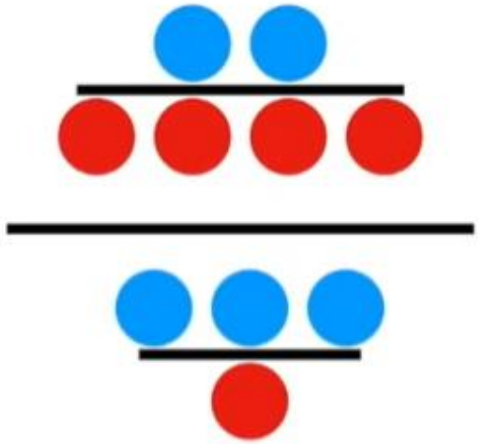


…to these odds.

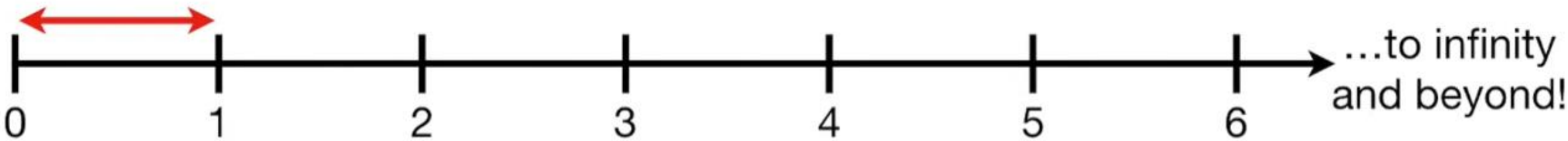


Doing the math gives us…

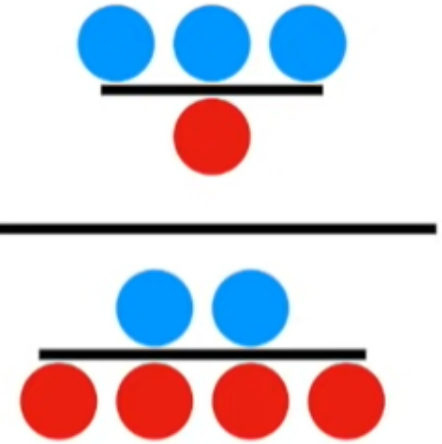
$$= \frac{2/4}{3/1}$$

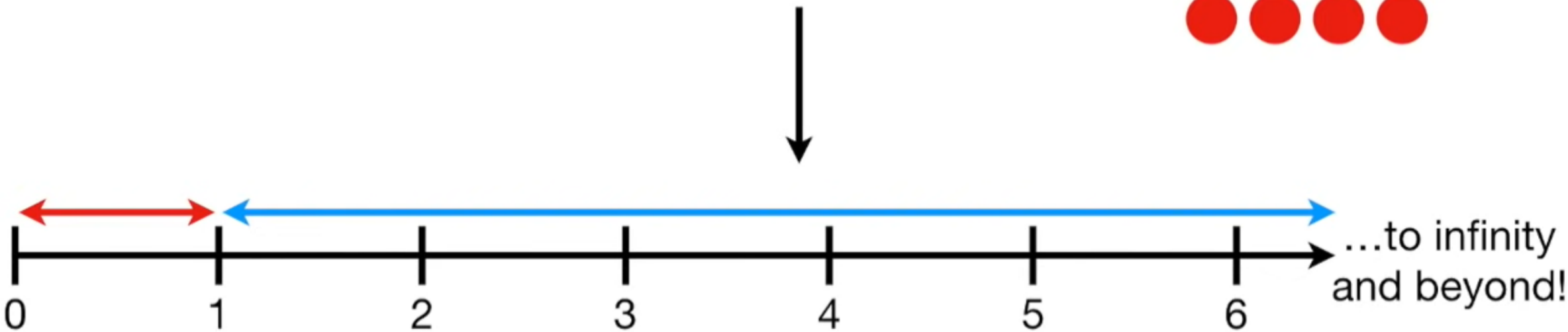


Just like when we calculate the odds of something, if the denominator is larger than the numerator, the odds ratio will go from 0 to 1…

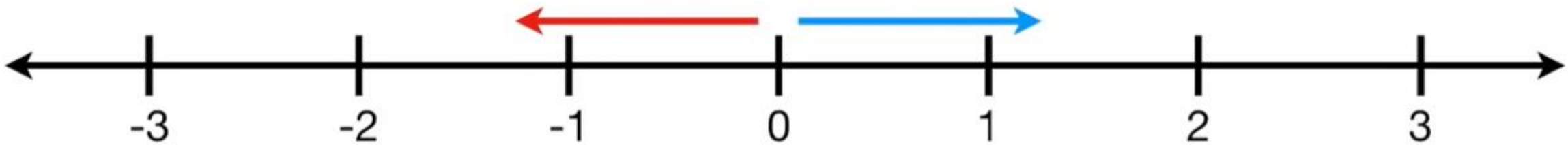


…and if the numerator is larger than the denominator, then the odds ratio will go from 1 to infinity (and beyond!)…

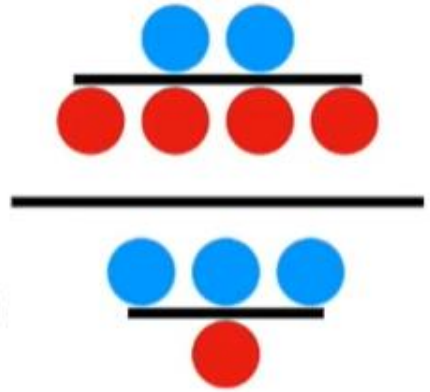




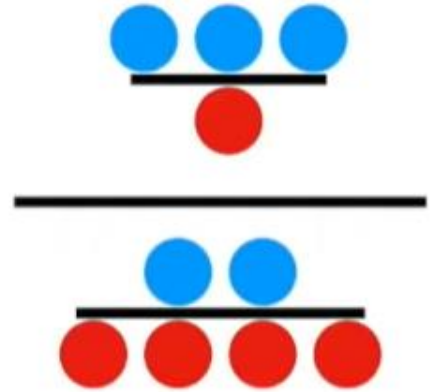
…and, just like the odds, taking the log of the odds ratio (i.e. log(odds ratio)) makes things nice and symmetrical.

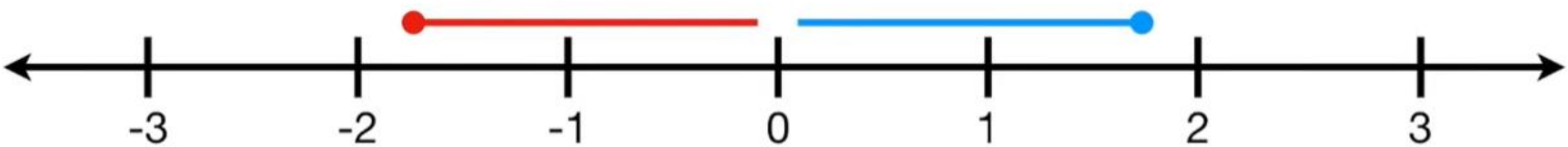


For example if the odds ratio is (2/4)/(3/1), then the log(odds ratio) = -1.79



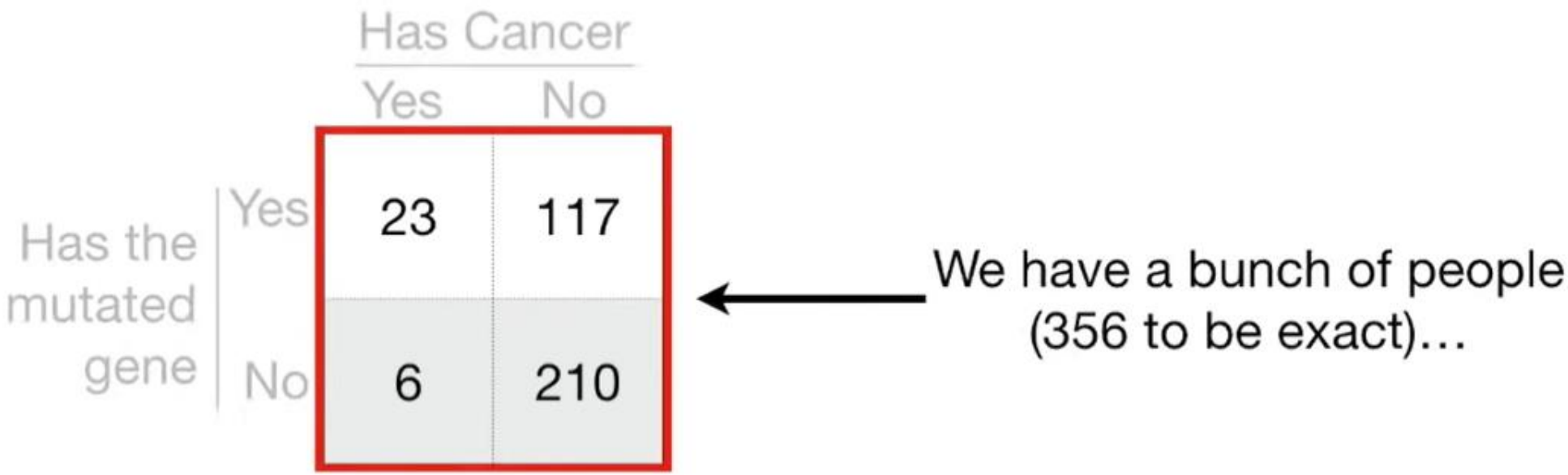
…and if the odds ratio is (3/1)/(2/4), then the log(odds ratio) = 1.79

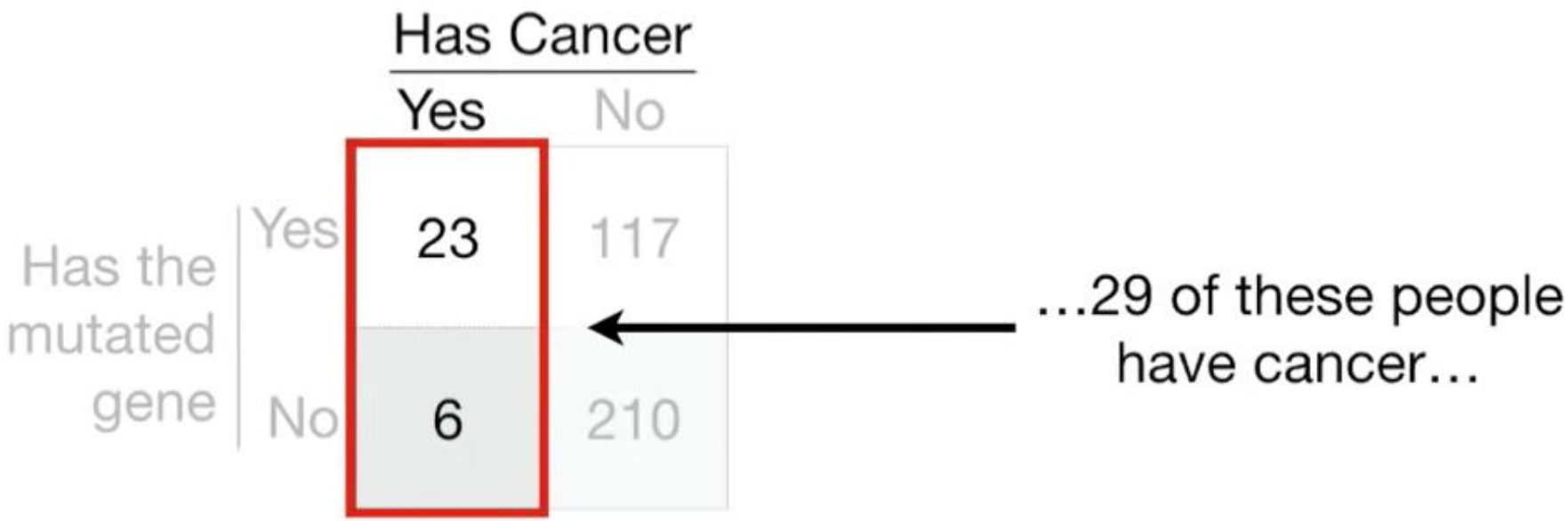


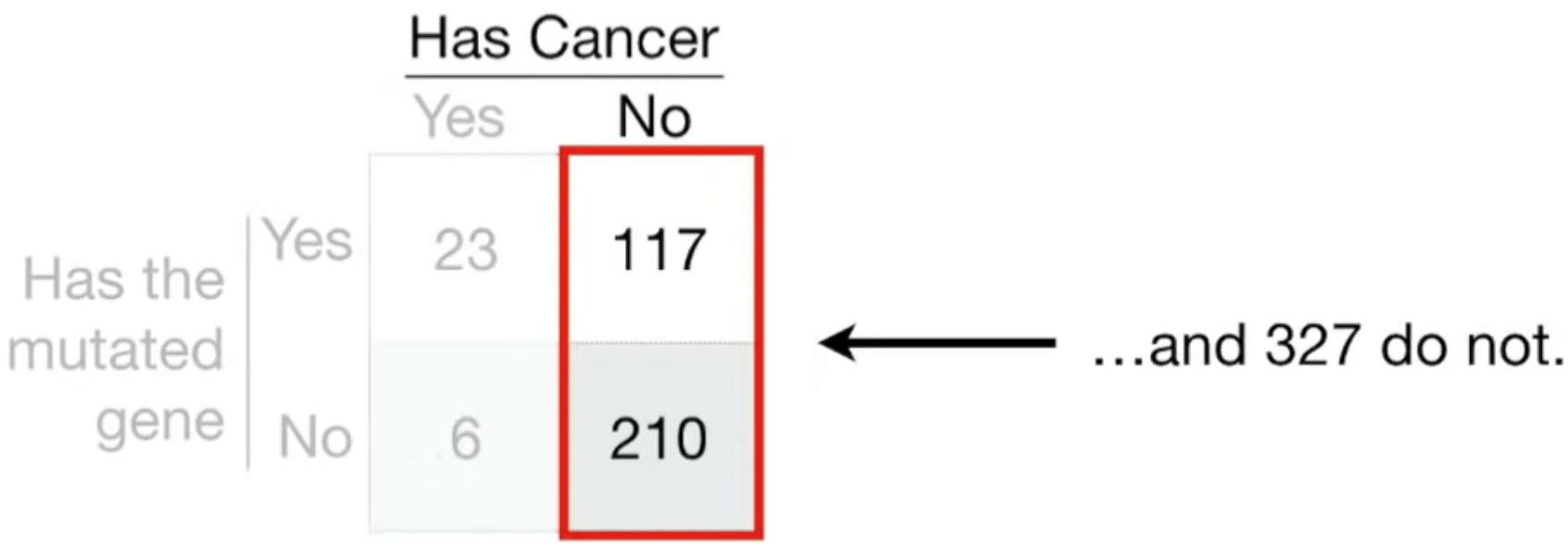


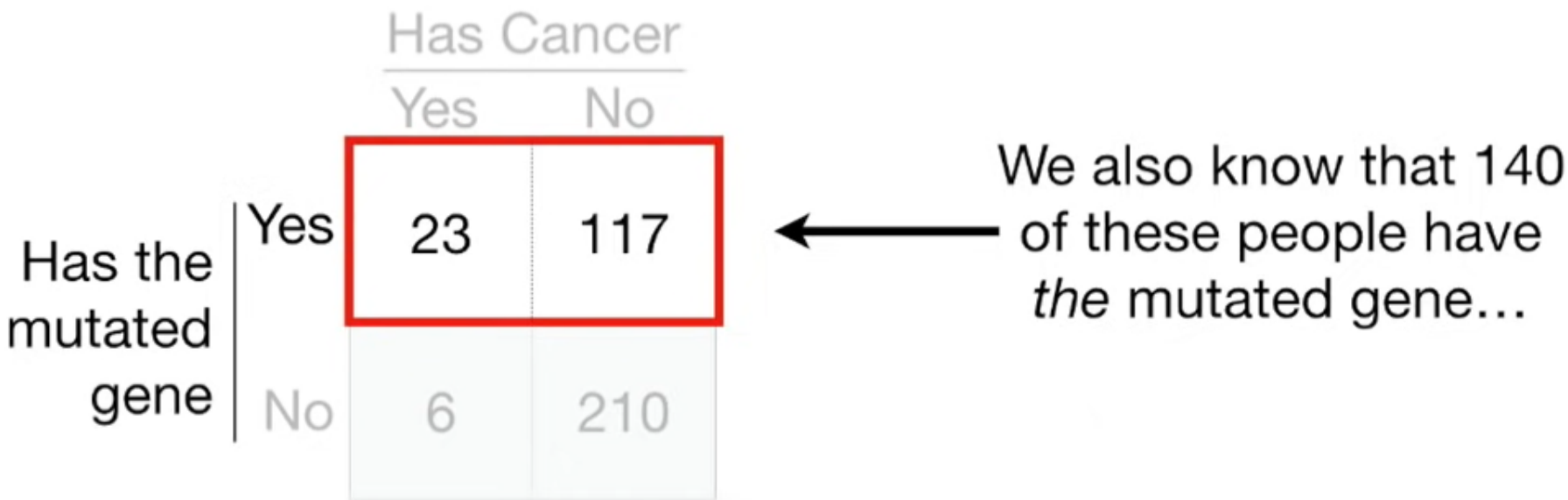
Here's an example of the "odds ratio" in action!

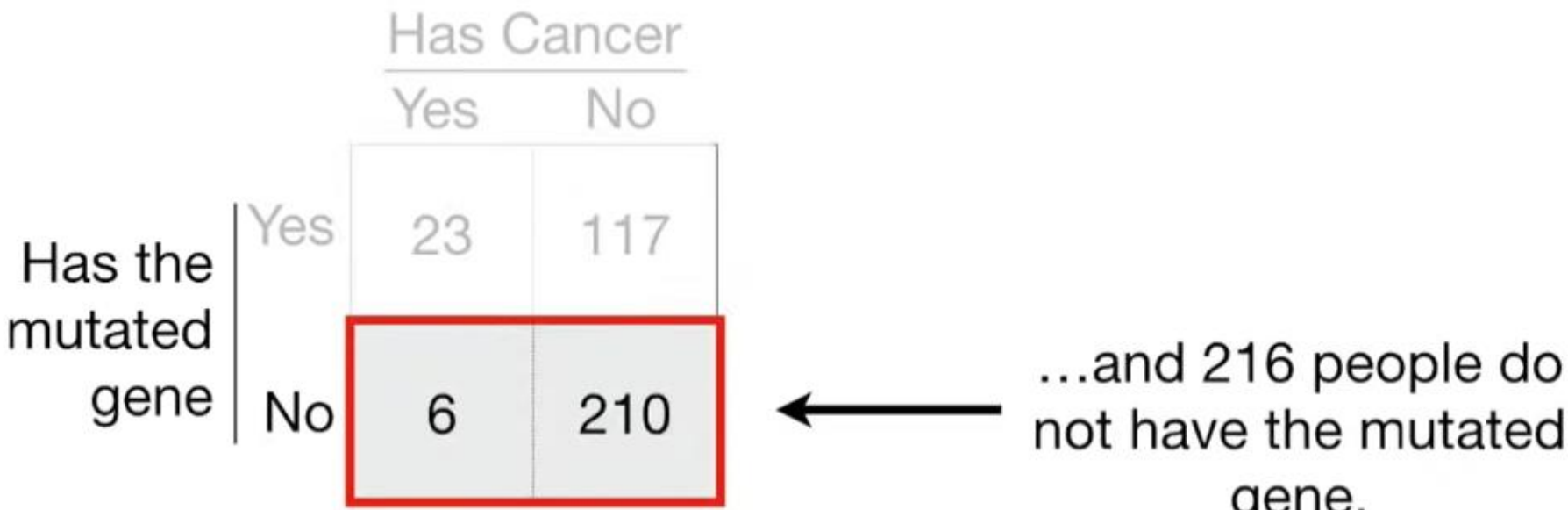
		Has Cancer: Yes	Has Cancer: No
Has the mutated gene	Yes	23	117
	No	6	210


Has Cancer
Yes
No
Has the mutated gene
Yes
No
23
117
6
210
We have a bunch of people (356 to be exact)…


Has Cancer
Yes
No
Has the mutated gene
Yes
No
23
117
6
210
…29 of these people have cancer…


Has Cancer
Yes
No
Has the mutated gene
Yes
No
23
117
6
210
…and 327 do not.


Has Cancer
Yes
No
Has the mutated gene
Yes
23
117
No
6
210
We also know that 140 of these people have *the* mutated gene…


Has Cancer
Yes
No
Has the mutated gene
Yes
23
117
No
6
210
…and 216 people do not have the mutated gene.

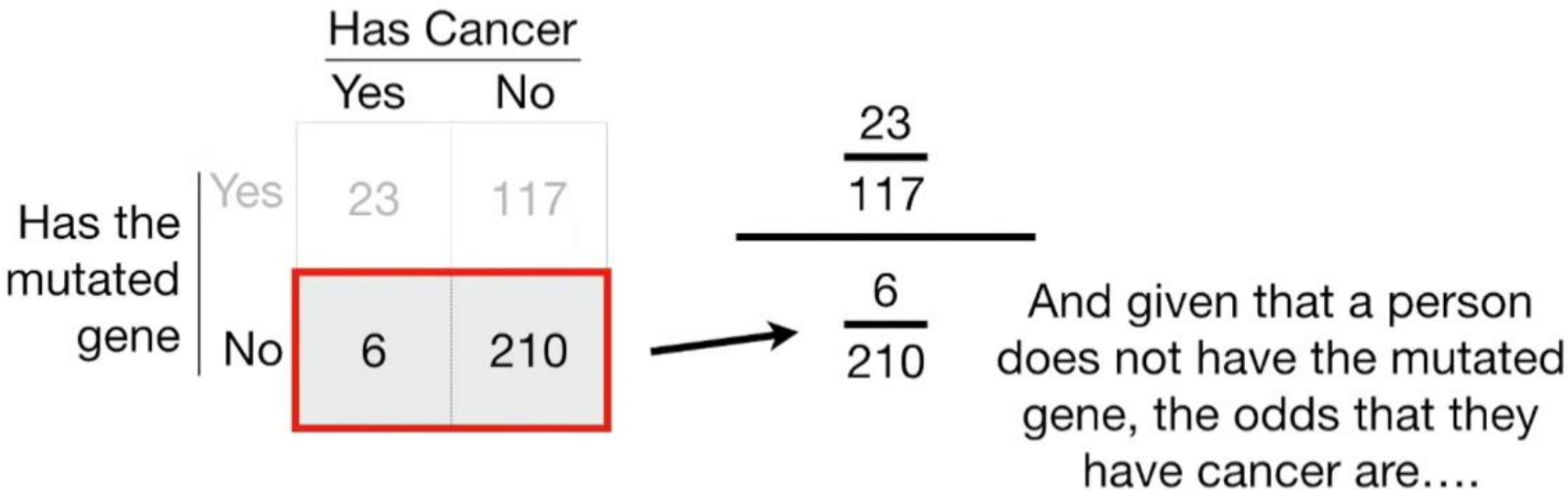
		Has Cancer: Yes	Has Cancer: No
Has the mutated gene	Yes	23	117
	No	6	210

We can use an “odds ratio” to determine if there is a relationship between the mutated gene and cancer.

Given that a person has has the mutated gene, the odds that they have cancer are….

Has the mutated gene	Has Cancer: Yes	Has Cancer: No
Yes	23	117
No	6	210

$$\frac{23}{117}$$


Has Cancer
Yes
No
Has the mutated gene
Yes
No
23
117
6
210
23/117
6/210
And given that a person does not have the mutated gene, the odds that they have cancer are….

		Has Cancer: Yes	Has Cancer: No
Has the mutated gene	Yes	23	117
	No	6	210

$$\frac{\frac{23}{117}}{\frac{6}{210}} = \frac{0.2}{0.03} = 6.88$$

…and the odds ratio tells us that the odds are 6.88 times greater that someone with the mutated gene will also have cancer.

		Has Cancer: Yes	Has Cancer: No
Has the mutated gene	Yes	23	117
	No	6	210

$$\frac{\frac{23}{117}}{\frac{6}{210}} = \frac{0.2}{0.03} = 6.88$$

$$\log(6.88) = 1.93$$

…and the log(odds ratio) is 1.93.

The odds ratio and the log(odds ratio) are like R-squared; they indicate a relationship between two things (in this case, a relationship between the mutated gene and cancer)…

		Has Cancer: Yes	Has Cancer: No
Has the mutated gene	Yes	23	117
	No	6	210

$$\frac{\frac{23}{117}}{\frac{6}{210}} = \frac{0.2}{0.03} = 6.88$$

$$\log(6.88) = 1.93$$

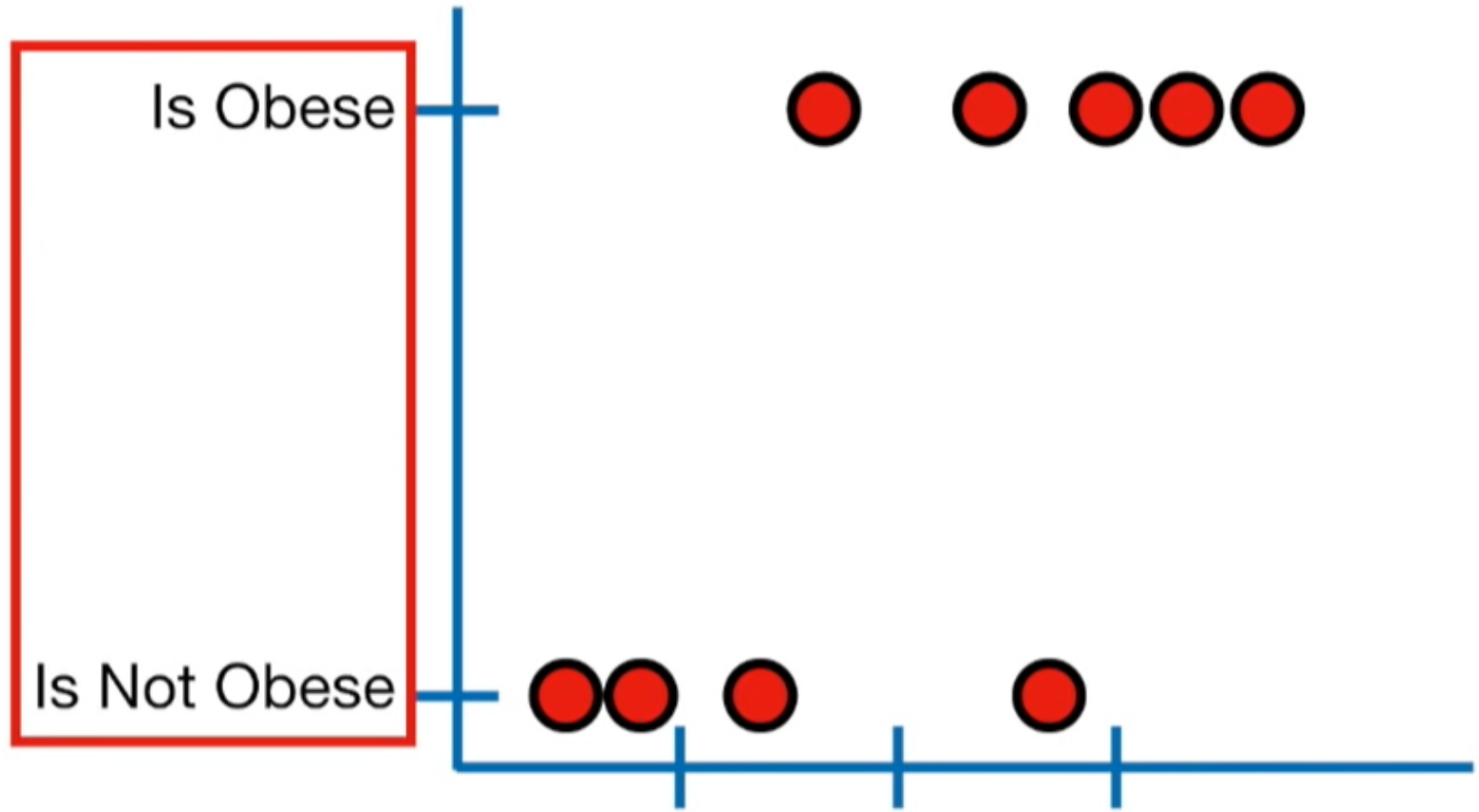
…larger values mean that the mutated gene is a good predictor of cancer. Smaller values mean that the mutated gene is not a good predictor of cancer.

		Has Cancer: Yes	Has Cancer: No
Has the mutated gene	Yes	23	117
	No	6	210

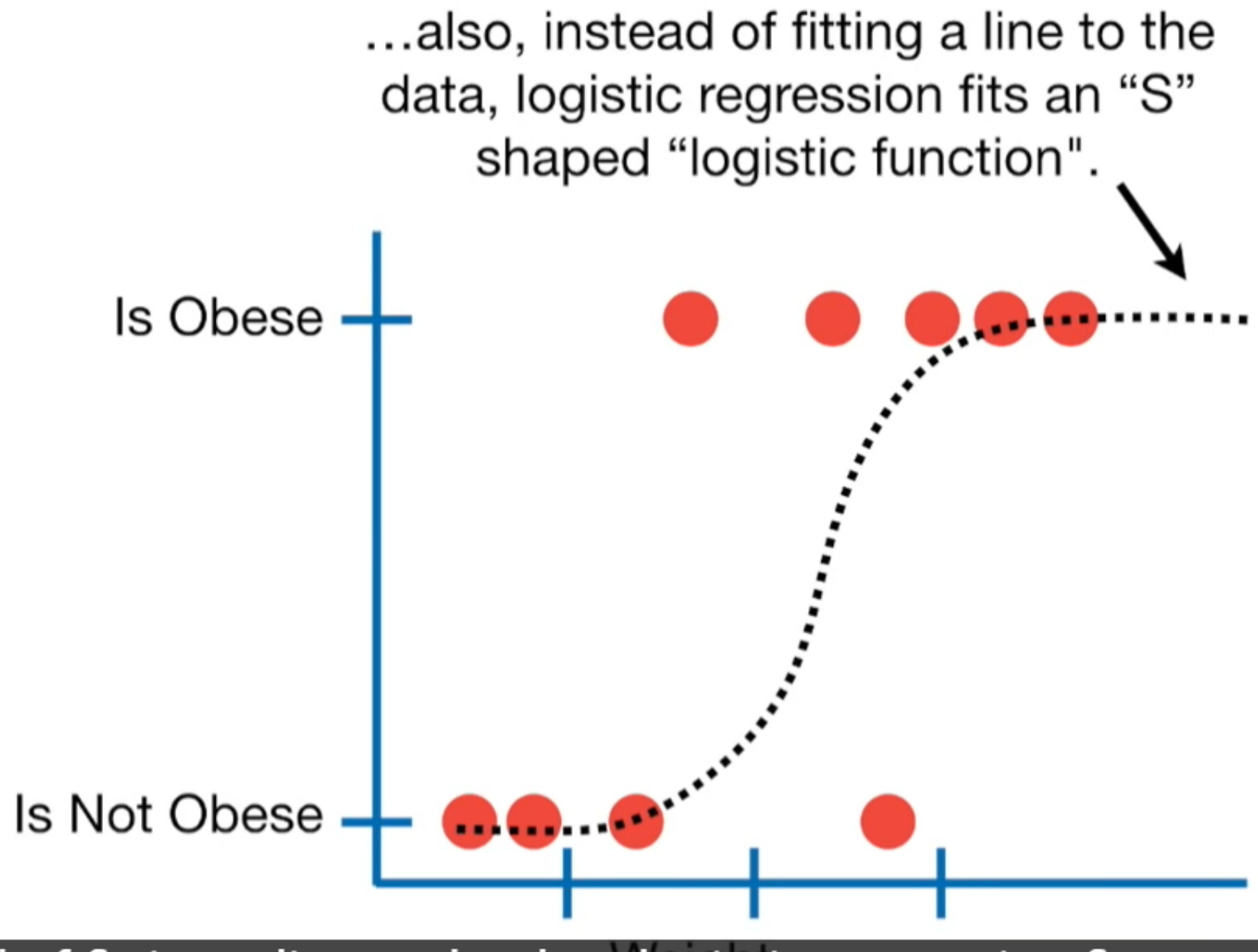
$$\frac{\frac{23}{117}}{\frac{6}{210}} = \frac{0.2}{0.03} = 6.88$$

$$\log(6.88) = 1.93$$

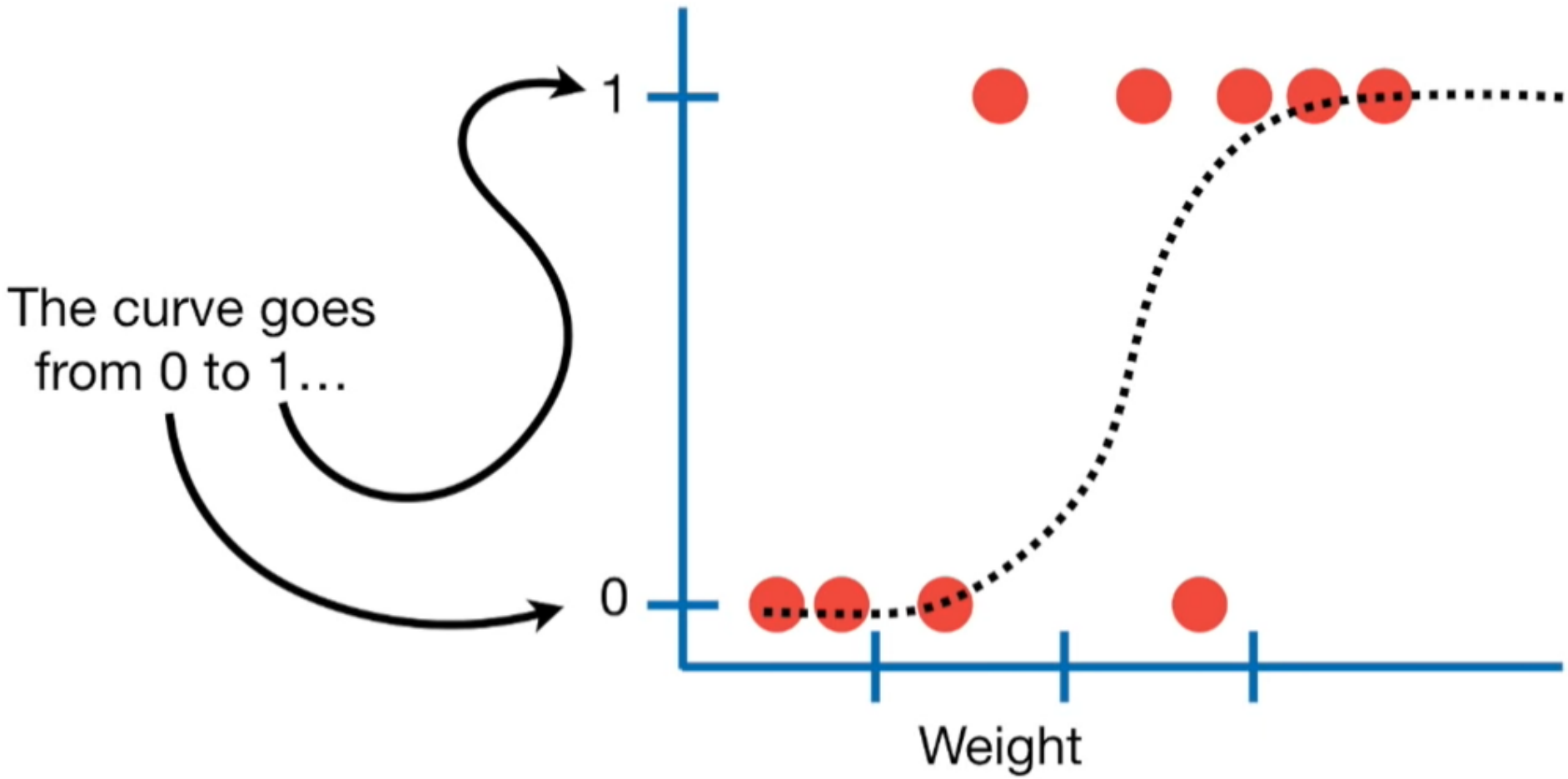
Logistic regression predicts whether something is ***True*** or ***False***, instead of predicting something continuous like ***size***.



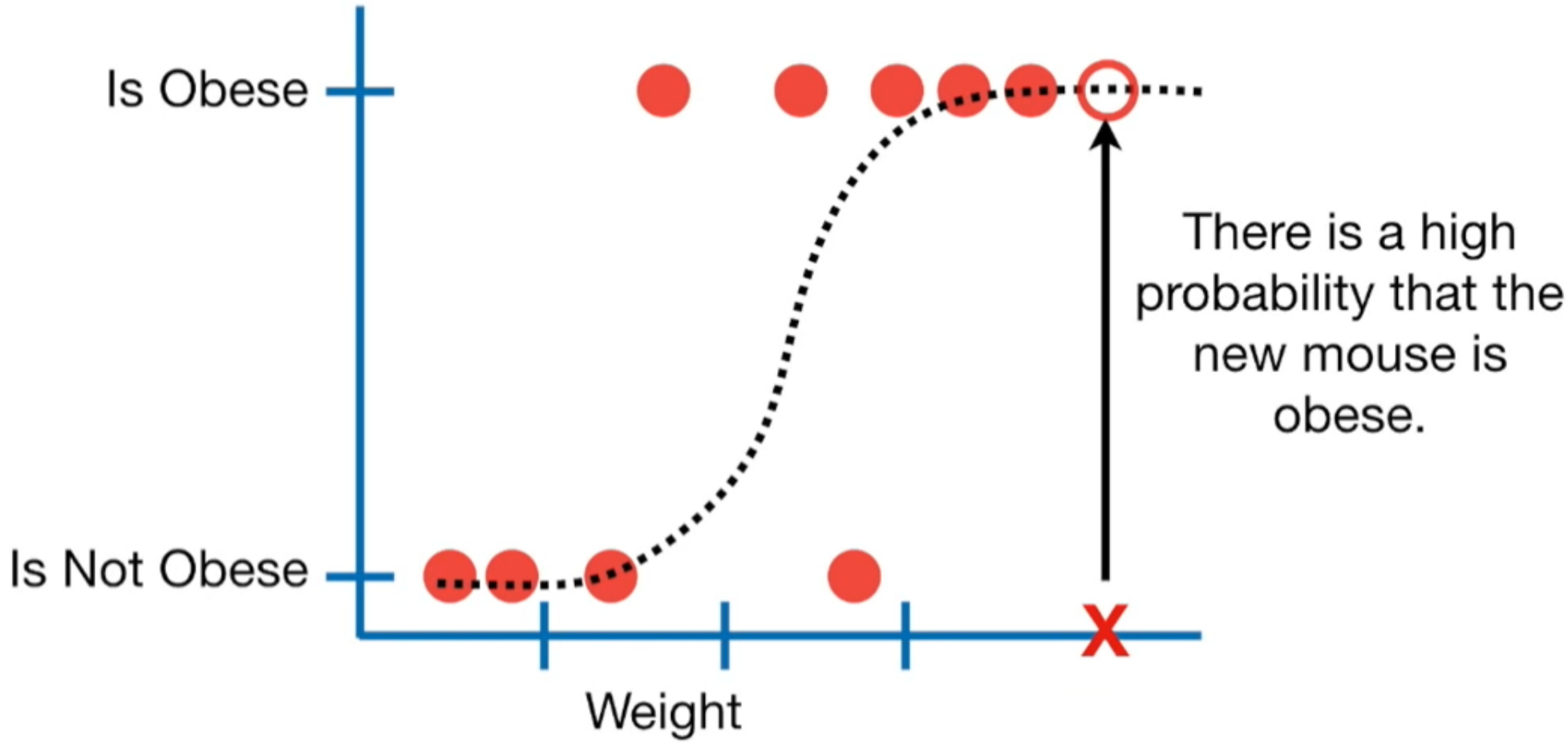
Logistic regression predicts whether something, is true or false instead of predicting something continuous, like, sighs



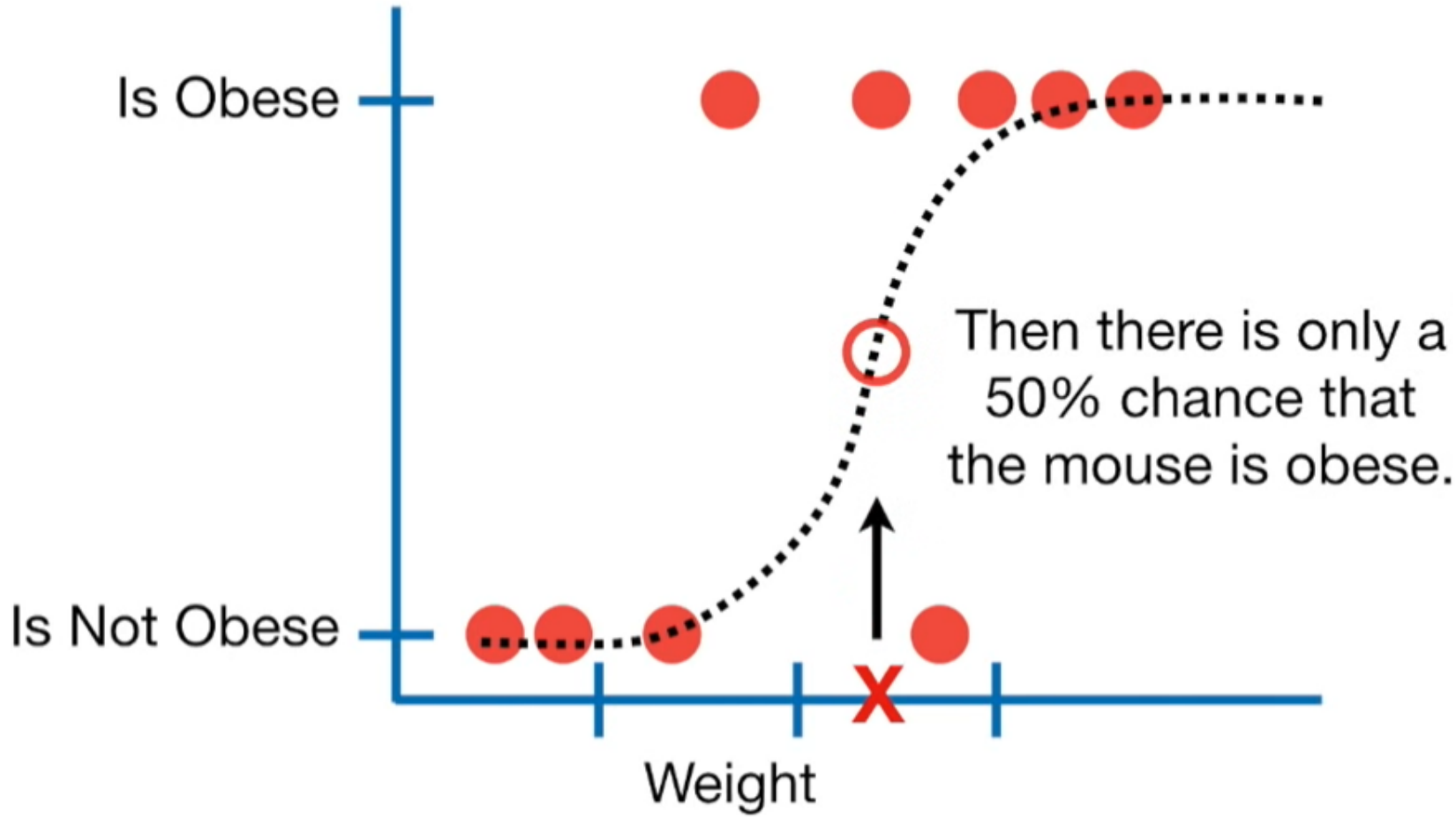
Also instead of fitting a line to the data logistic regression fits an s-shaped logistic function



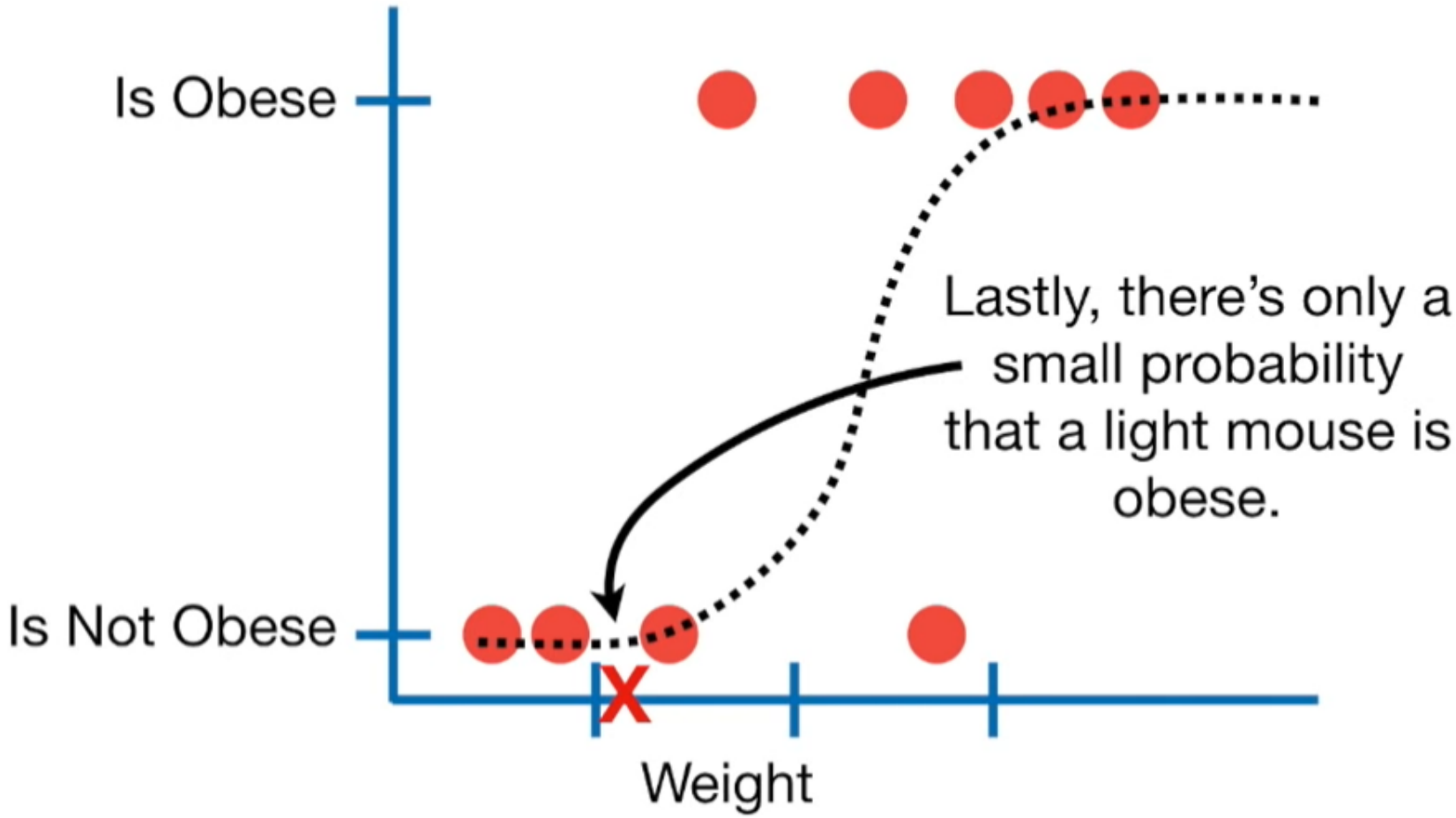
The curve goes from zero to one?



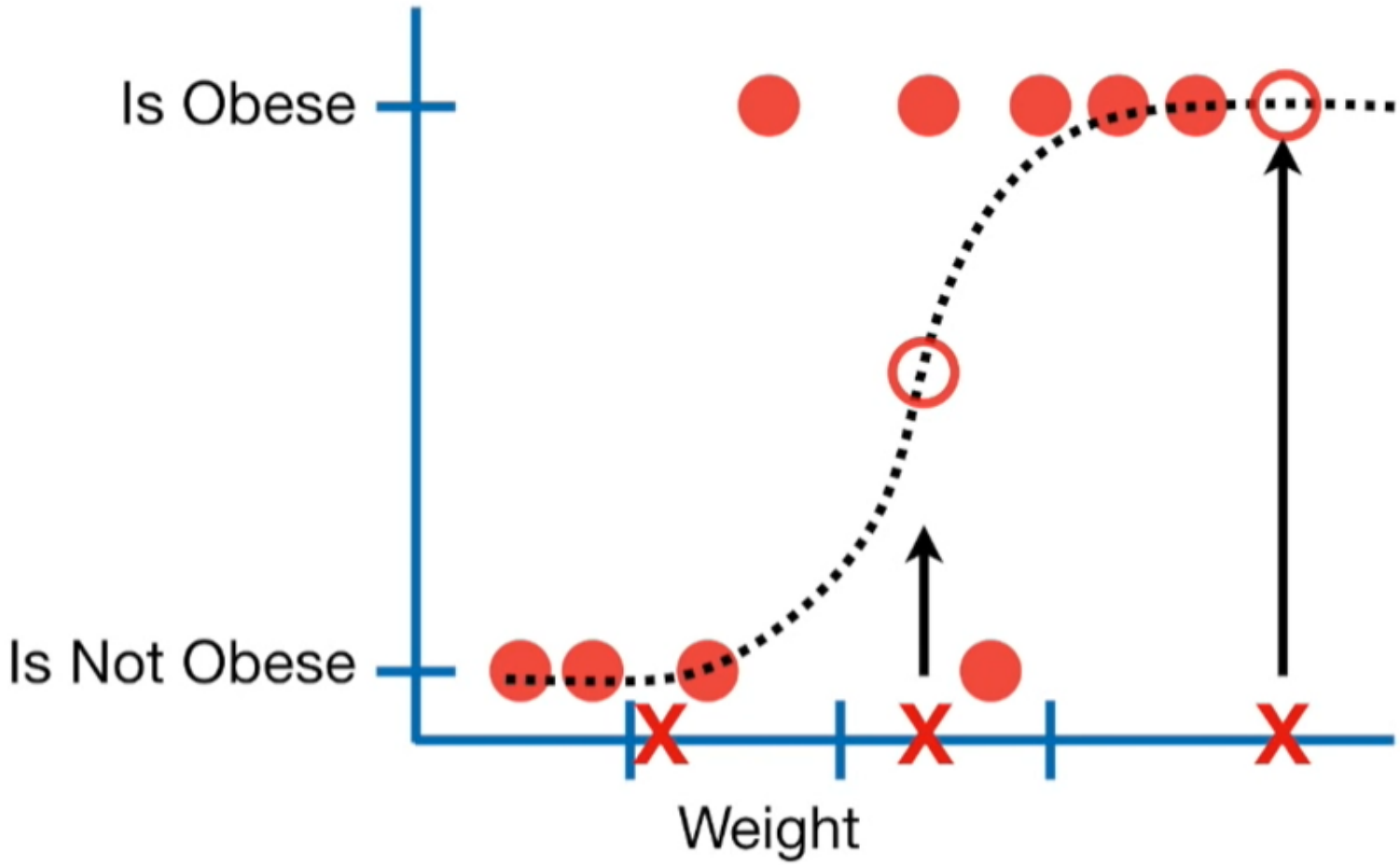
There is a high probability that the new, mouse is obese?

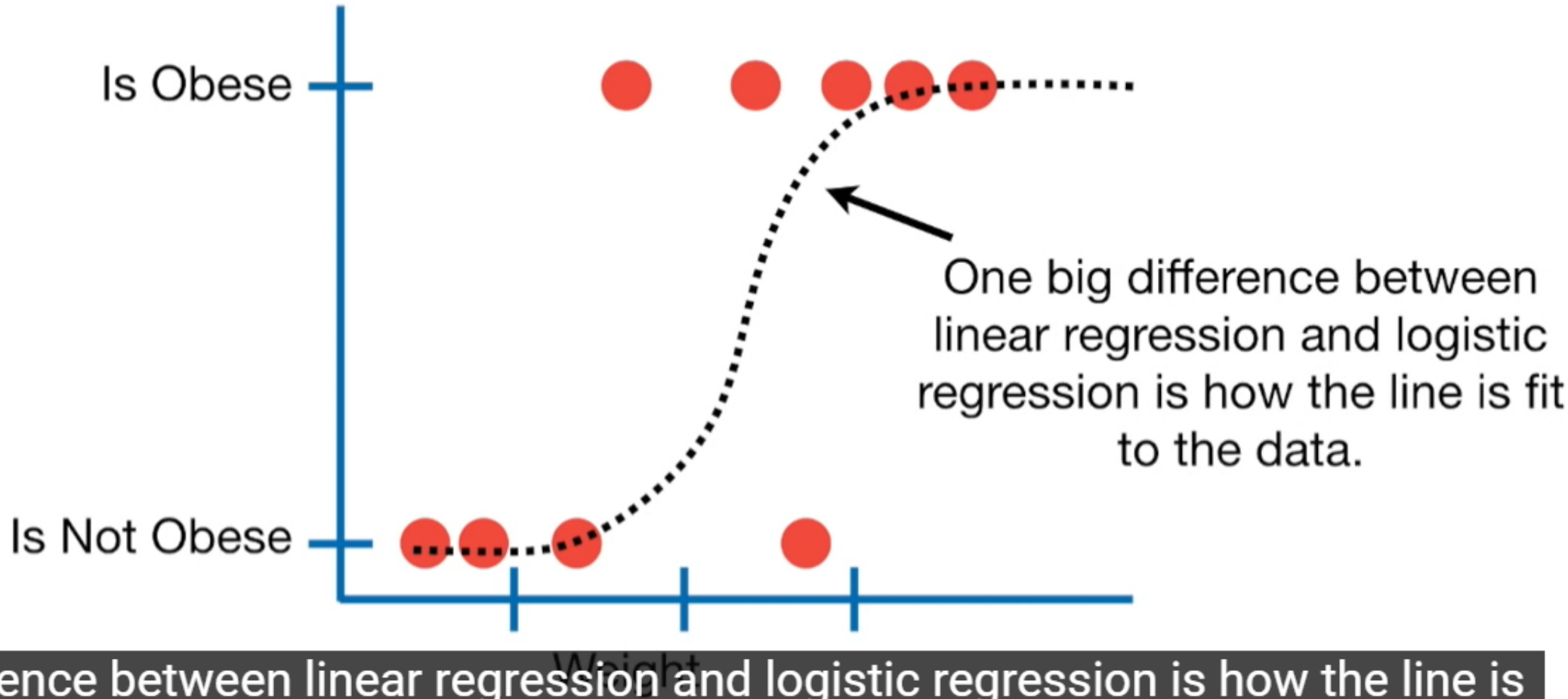


Then there is only a 50% chance of the mouse is obese?

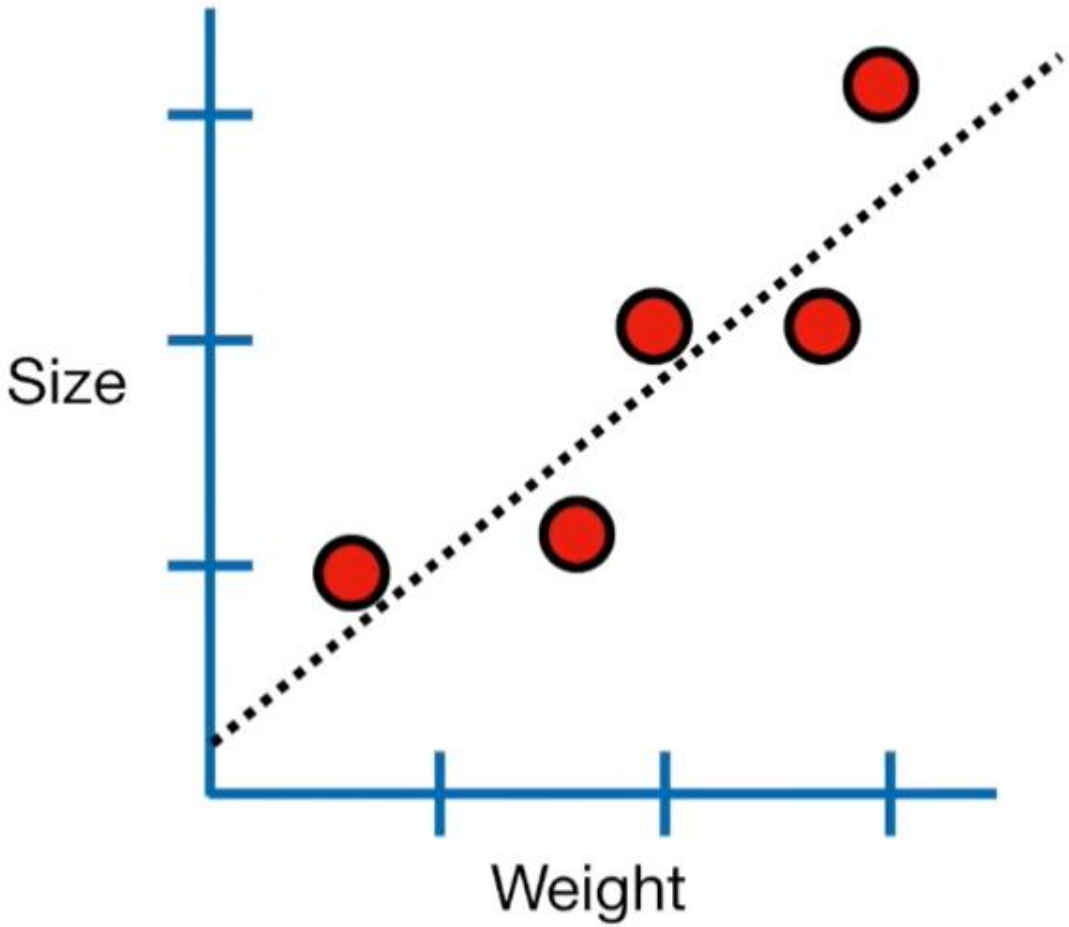

Is Obese
Is Not Obese
Lastly, there's only a small probability that a light mouse is obese.
X
Weight

For example, if the probability a mouse is obese is > 50%, then we'll classify it as obese, otherwise we'll classify it as "not obese".



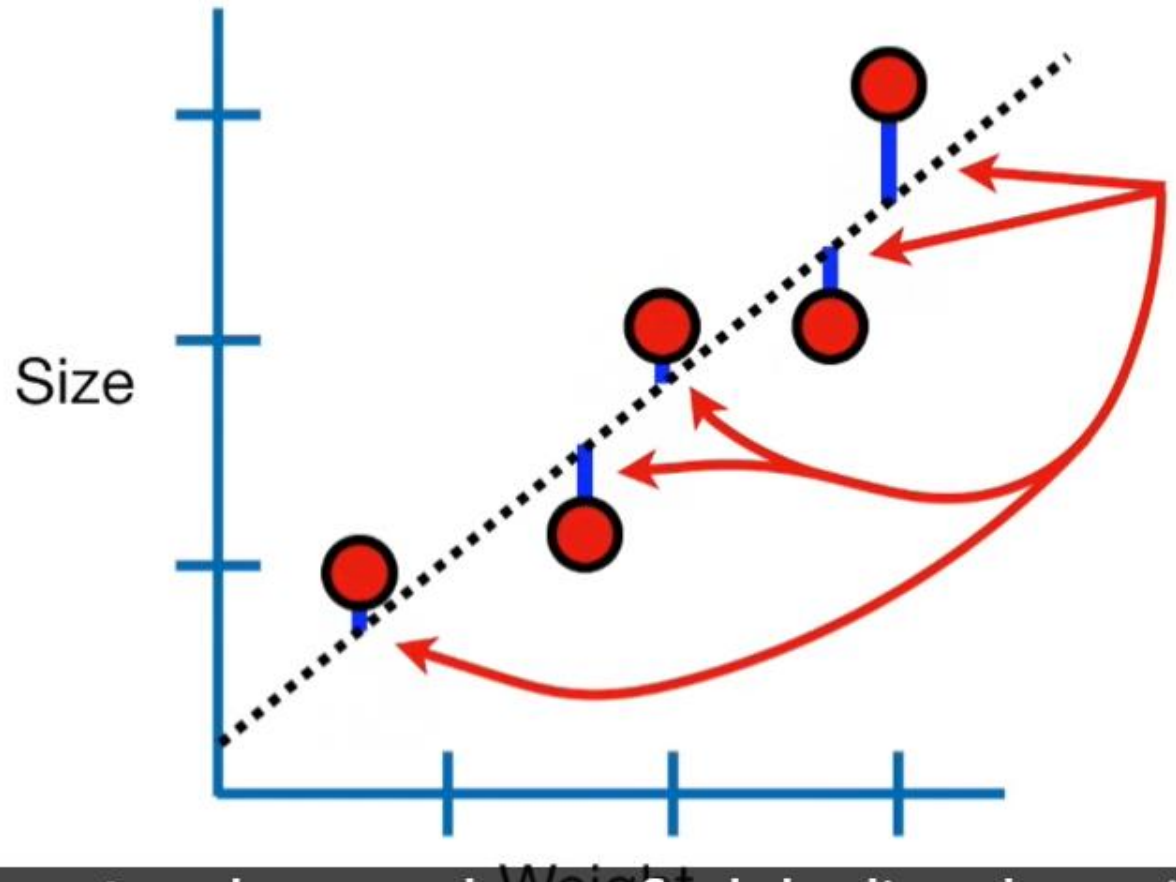


One big difference between linear regression and logistic regression is how the line is fit to the data



With linear regression, we fit the line using “least squares”.

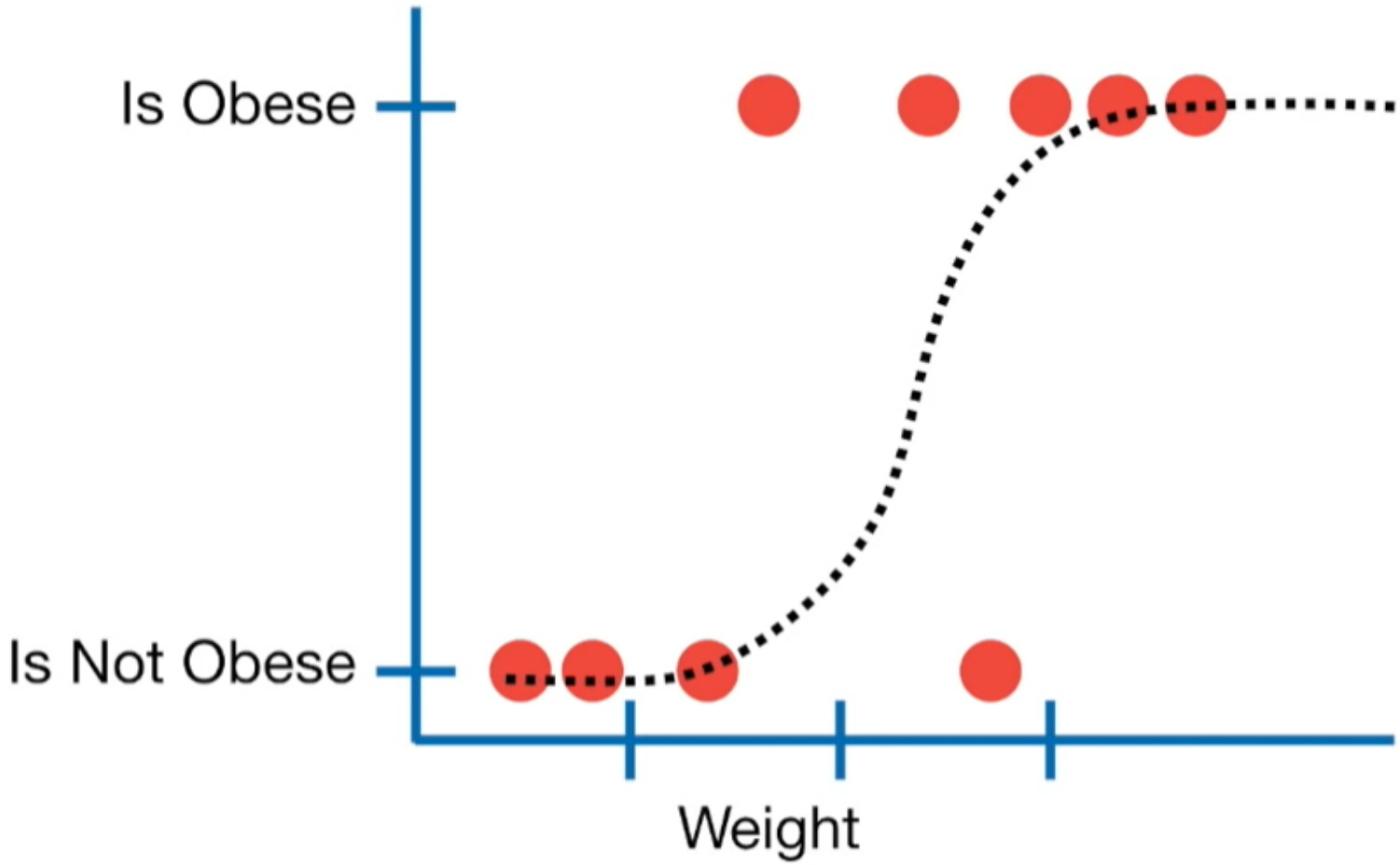
With linear regression, we fit the line, using least squares



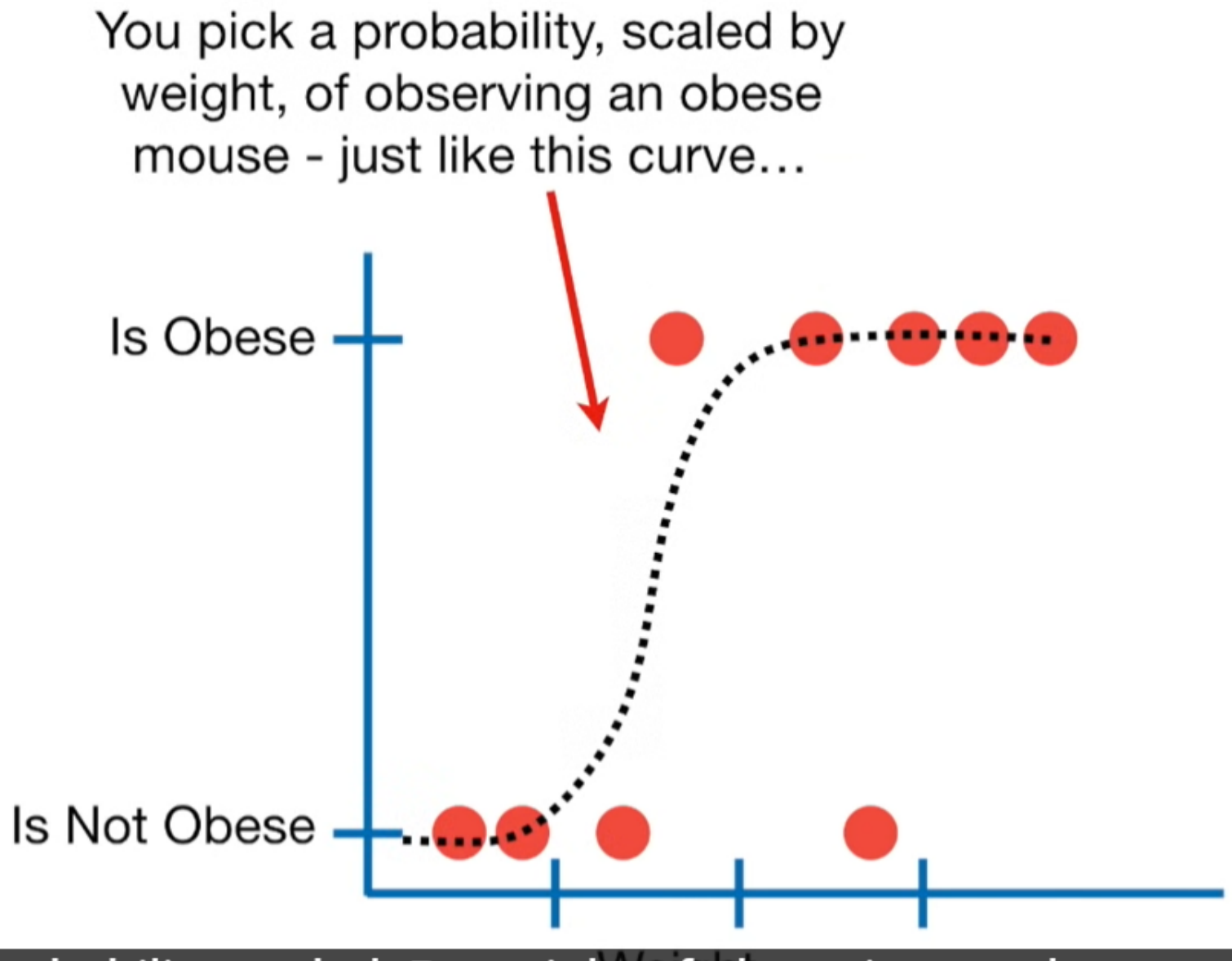
In other words, we find the line that minimizes the sum of the squares of these residuals.

In other words, we find the line that minimizes the sum of the squares of these residuals

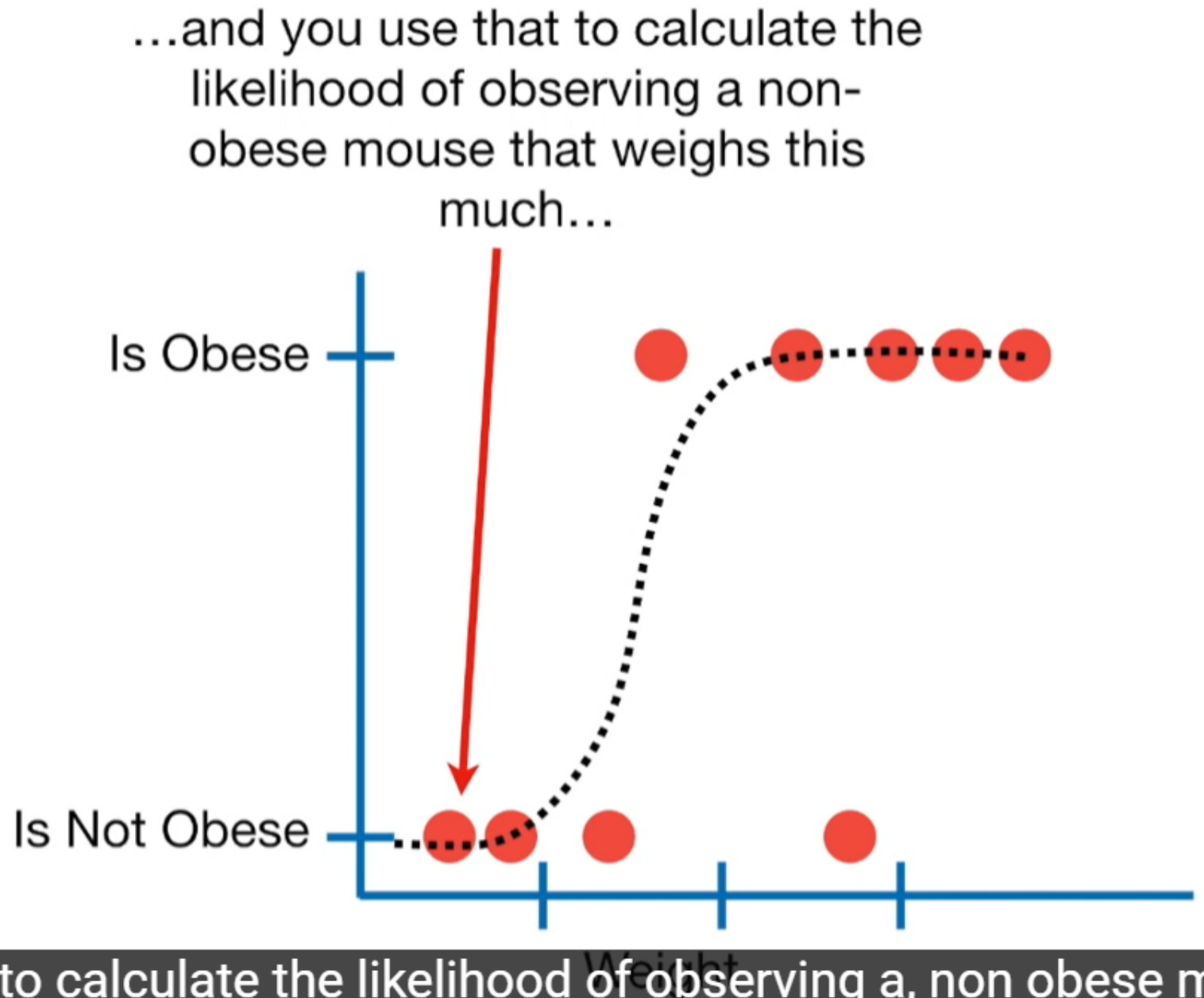
Instead it uses something called
“maximum likelihood”.



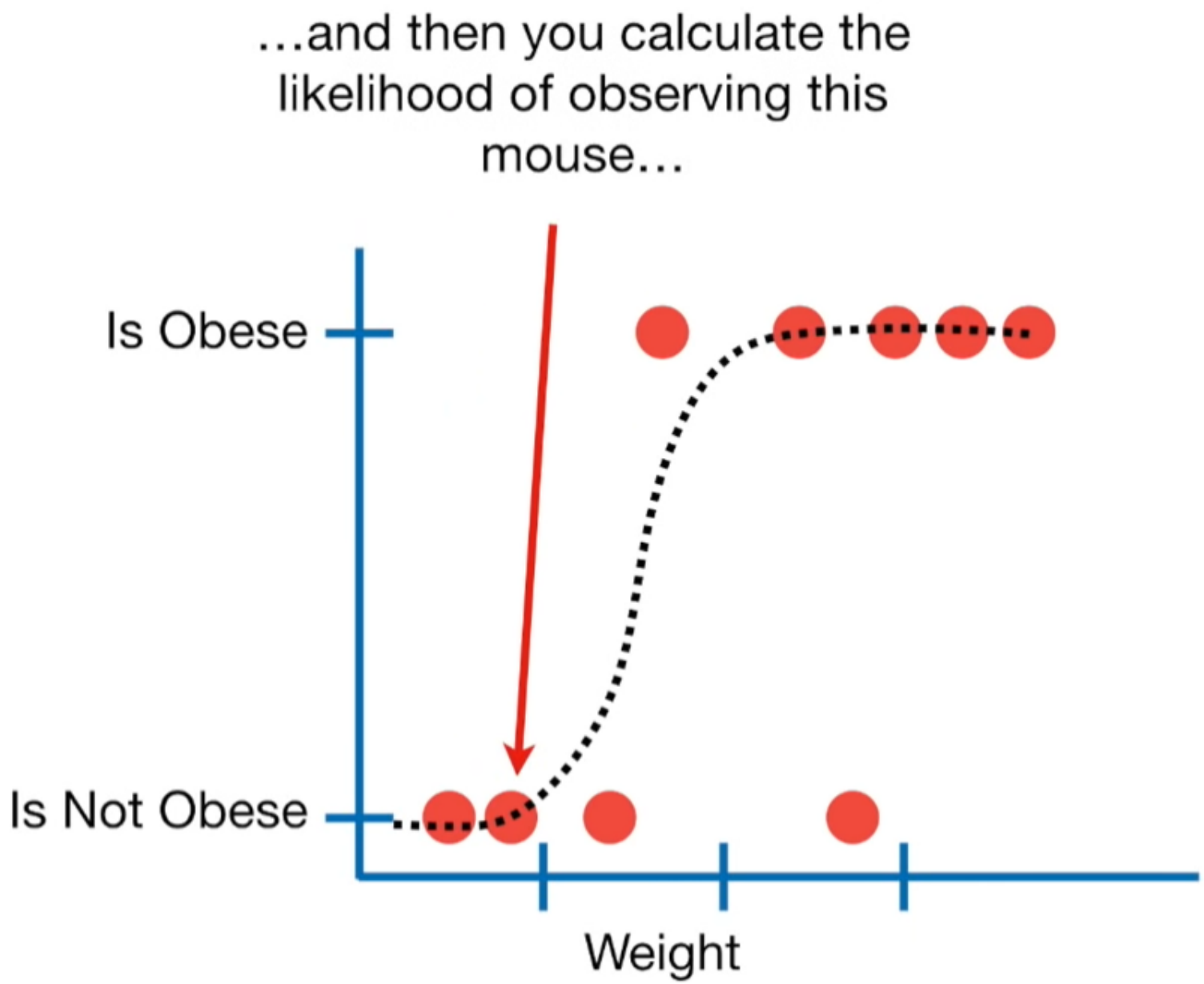
instead it uses something called maximum likelihood



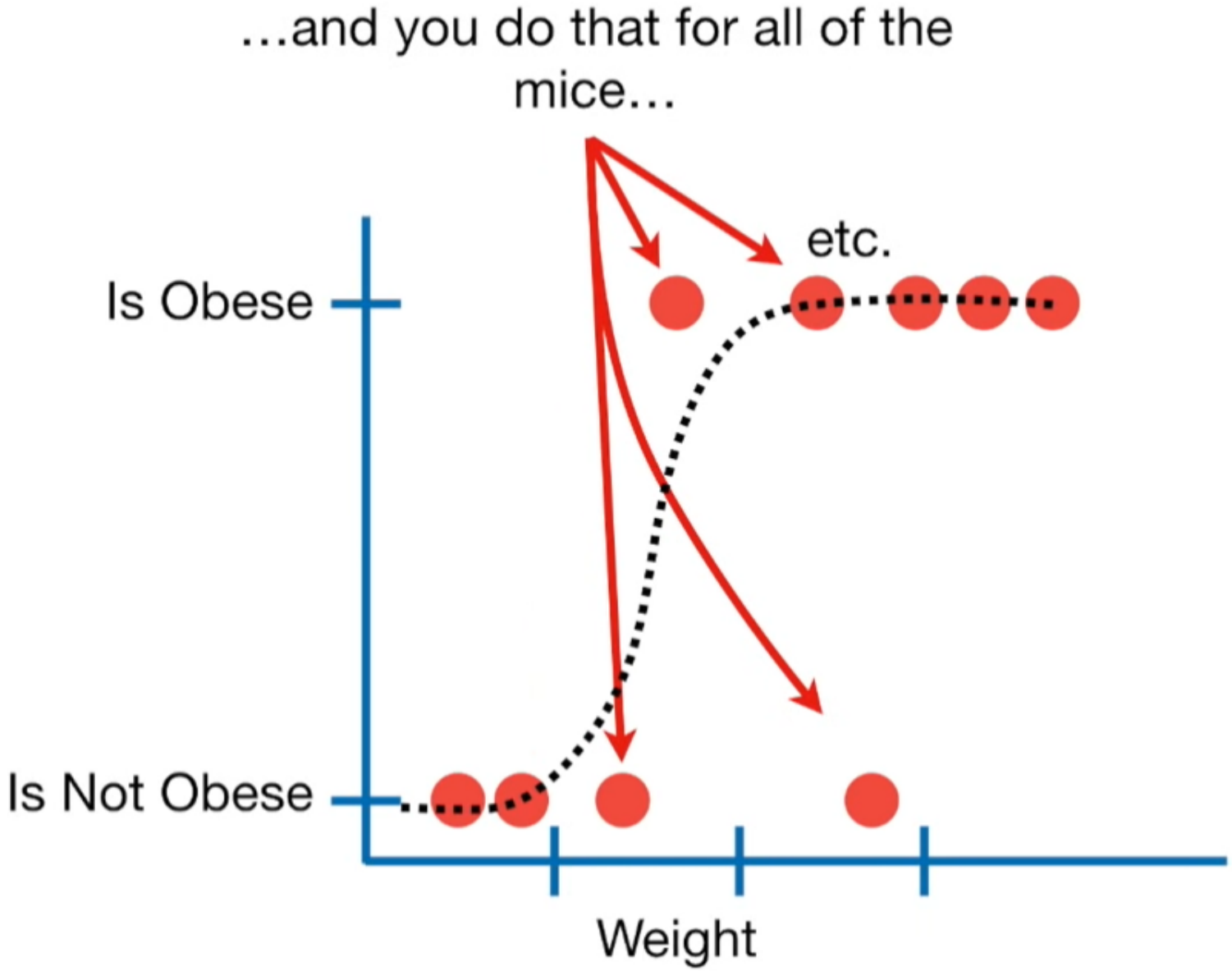
You, pick a probability scaled. By weight of observing an obese mouse just like this curve and



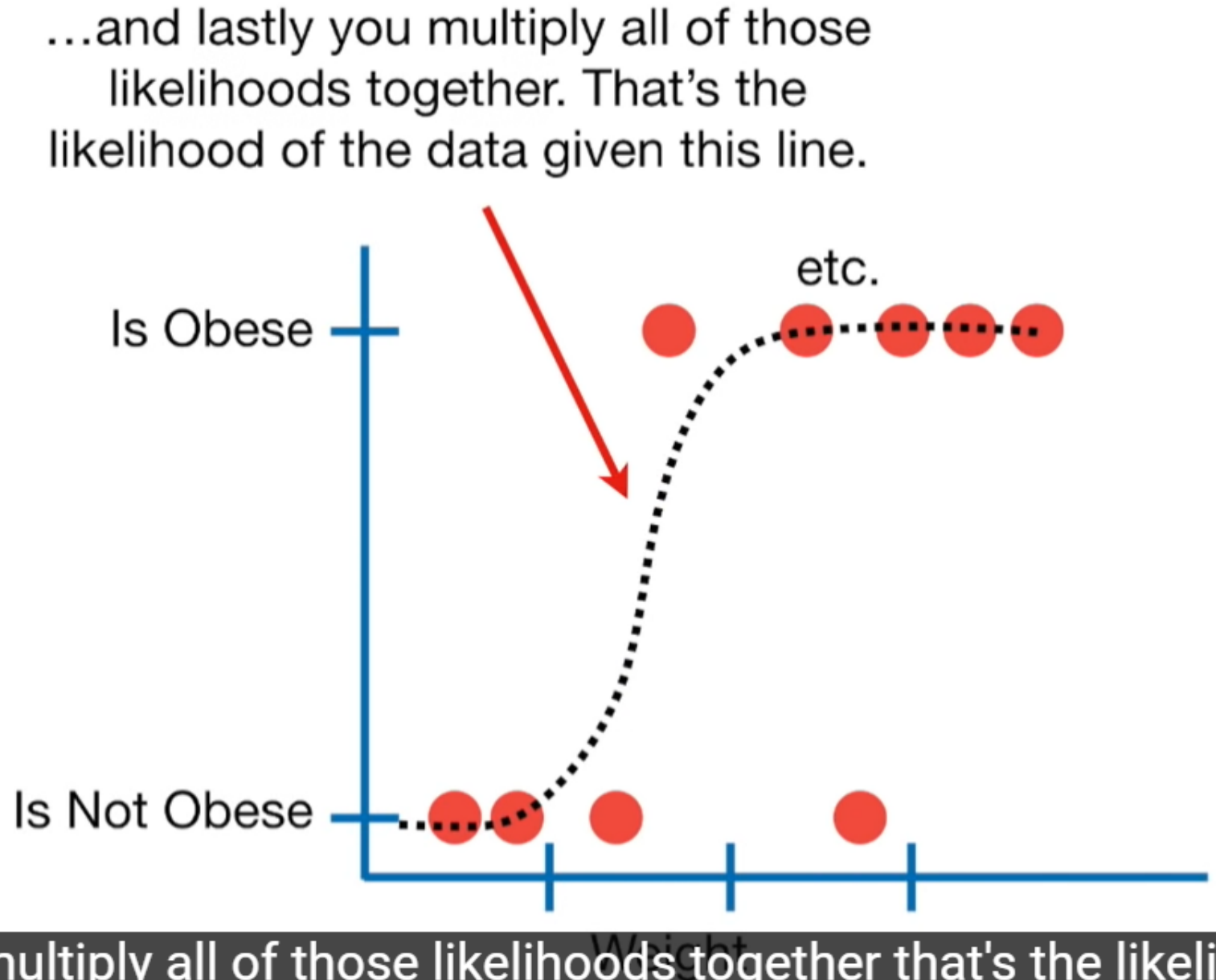
You, use that to calculate the likelihood of observing a, non obese mouse that weighs this much and



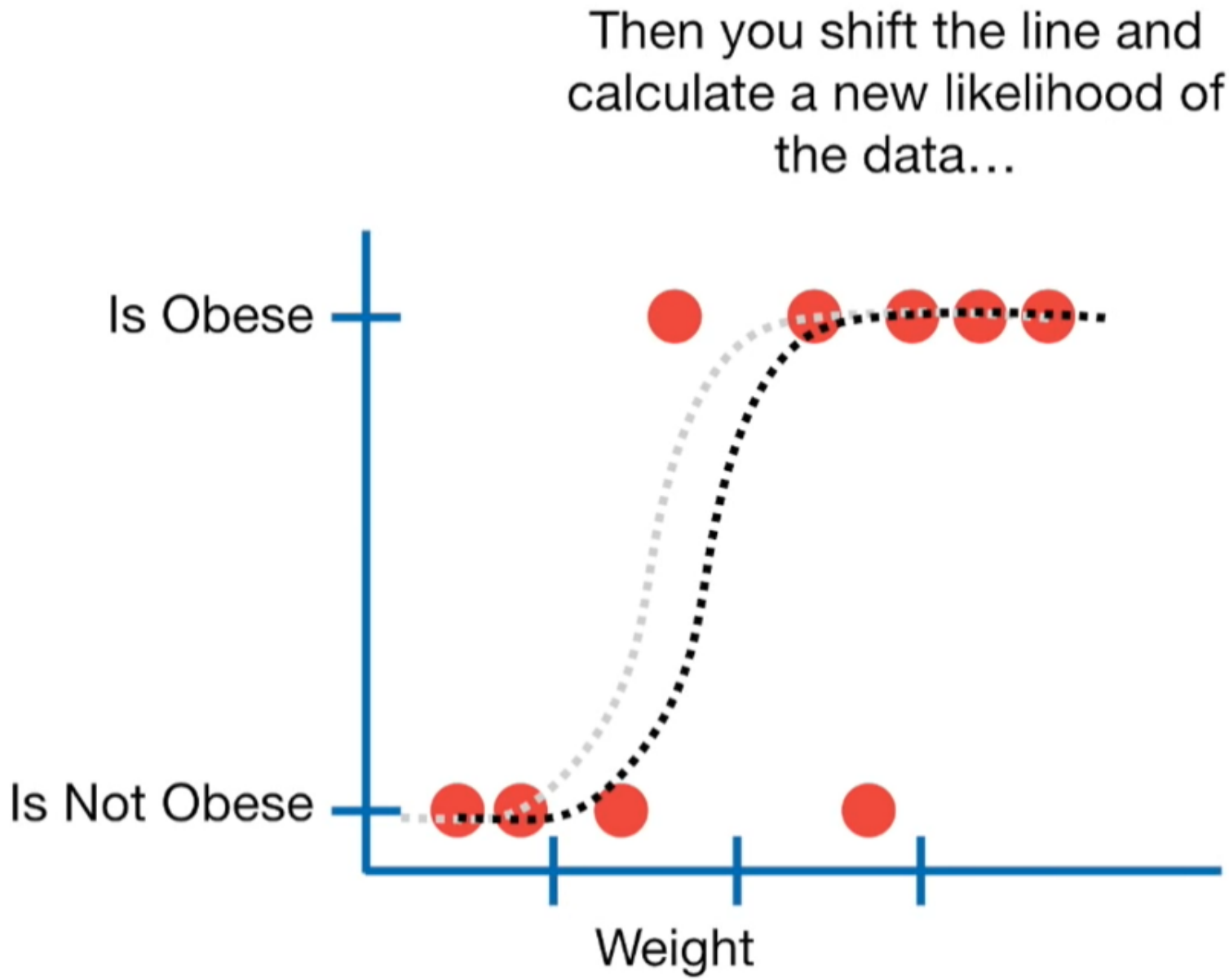
then you calculate the likelihood of observing, this mouse and



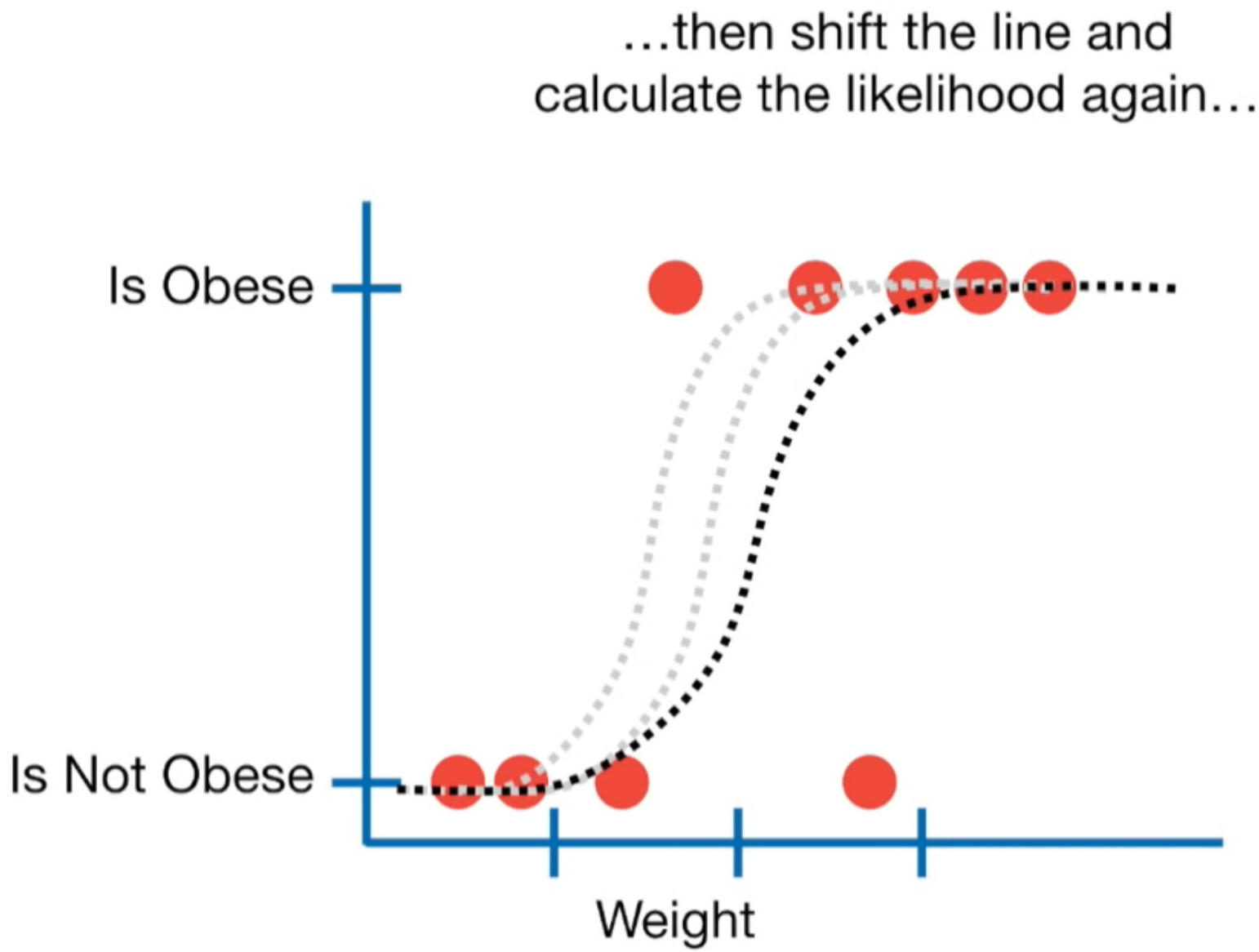
you, do that for all of the mice and



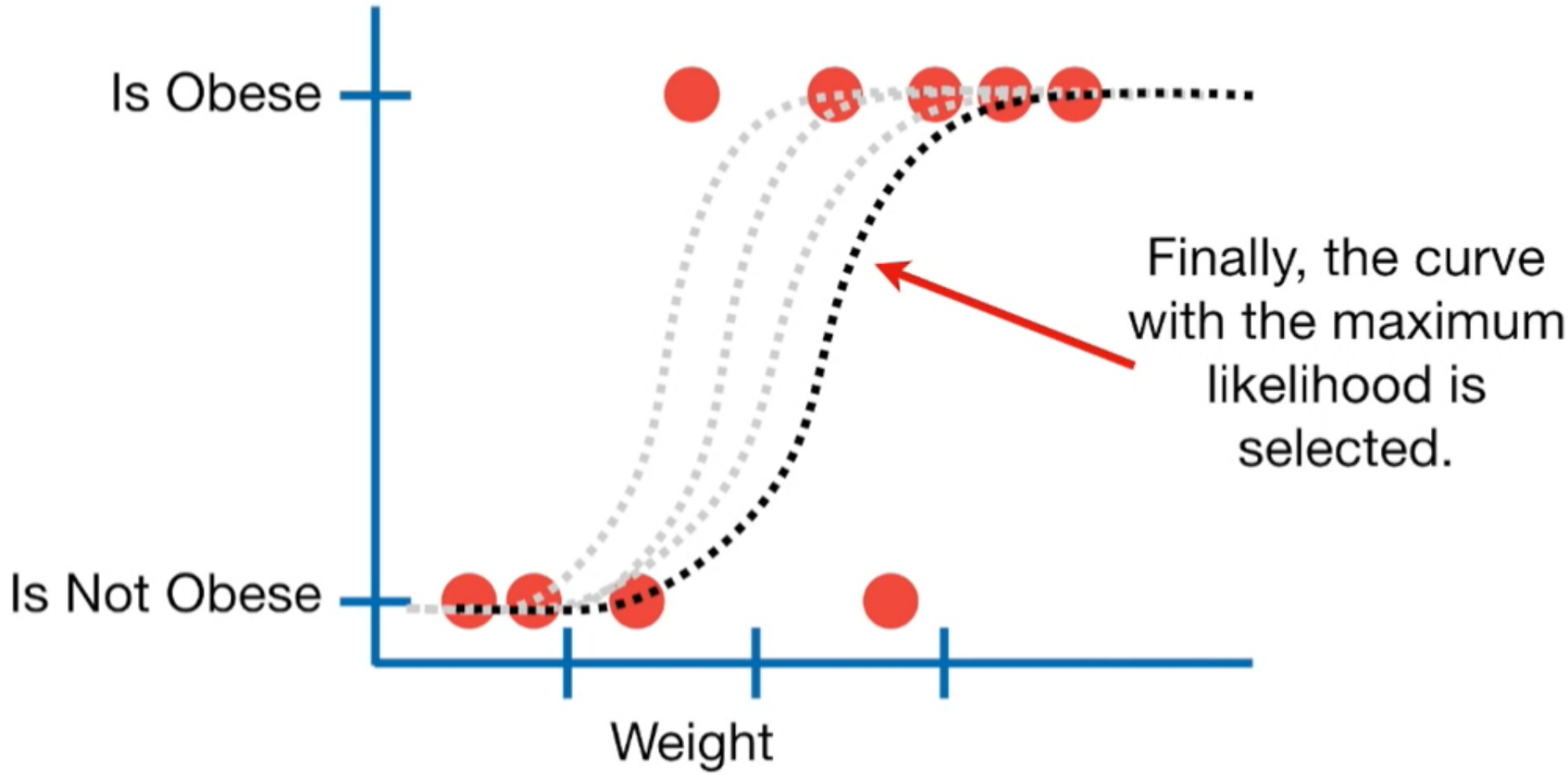
Lastly, you multiply all of those likelihoods together that's the likelihood of the data given this line



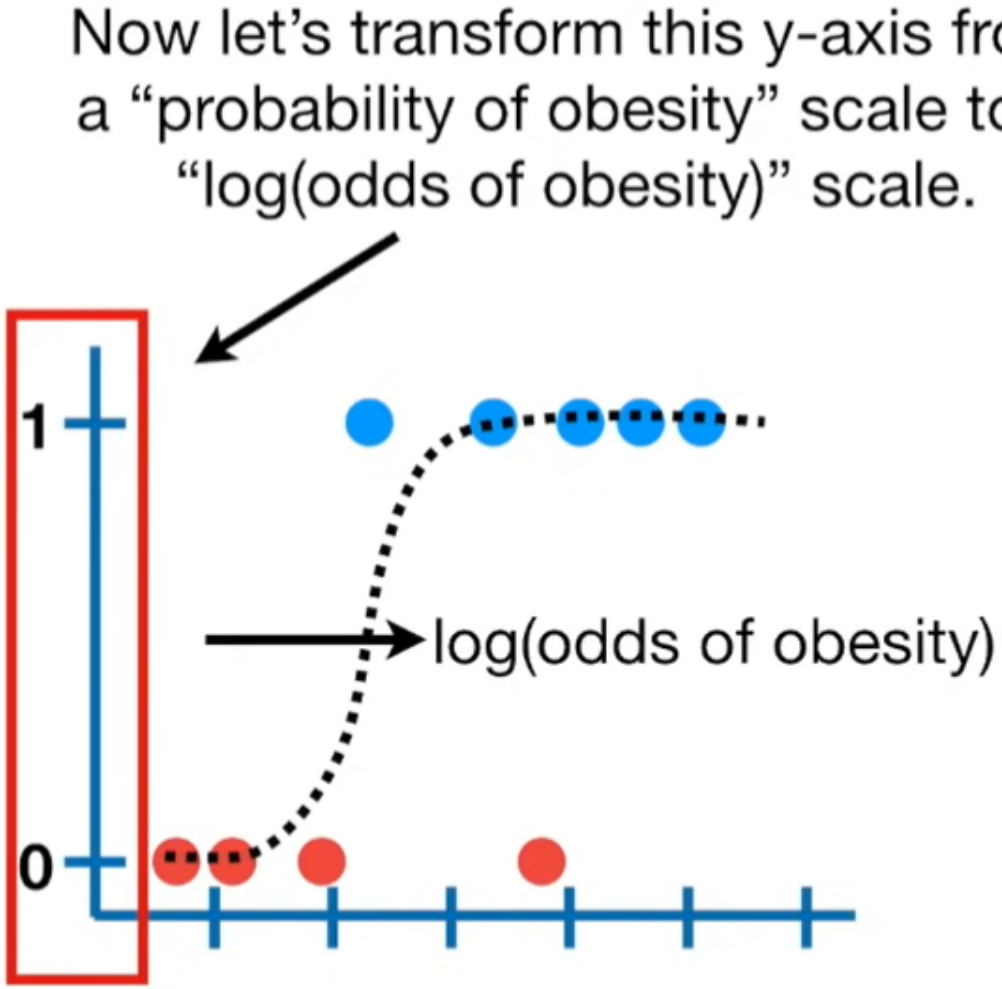
then you shift the line and calculate a new, likelihood of the data and



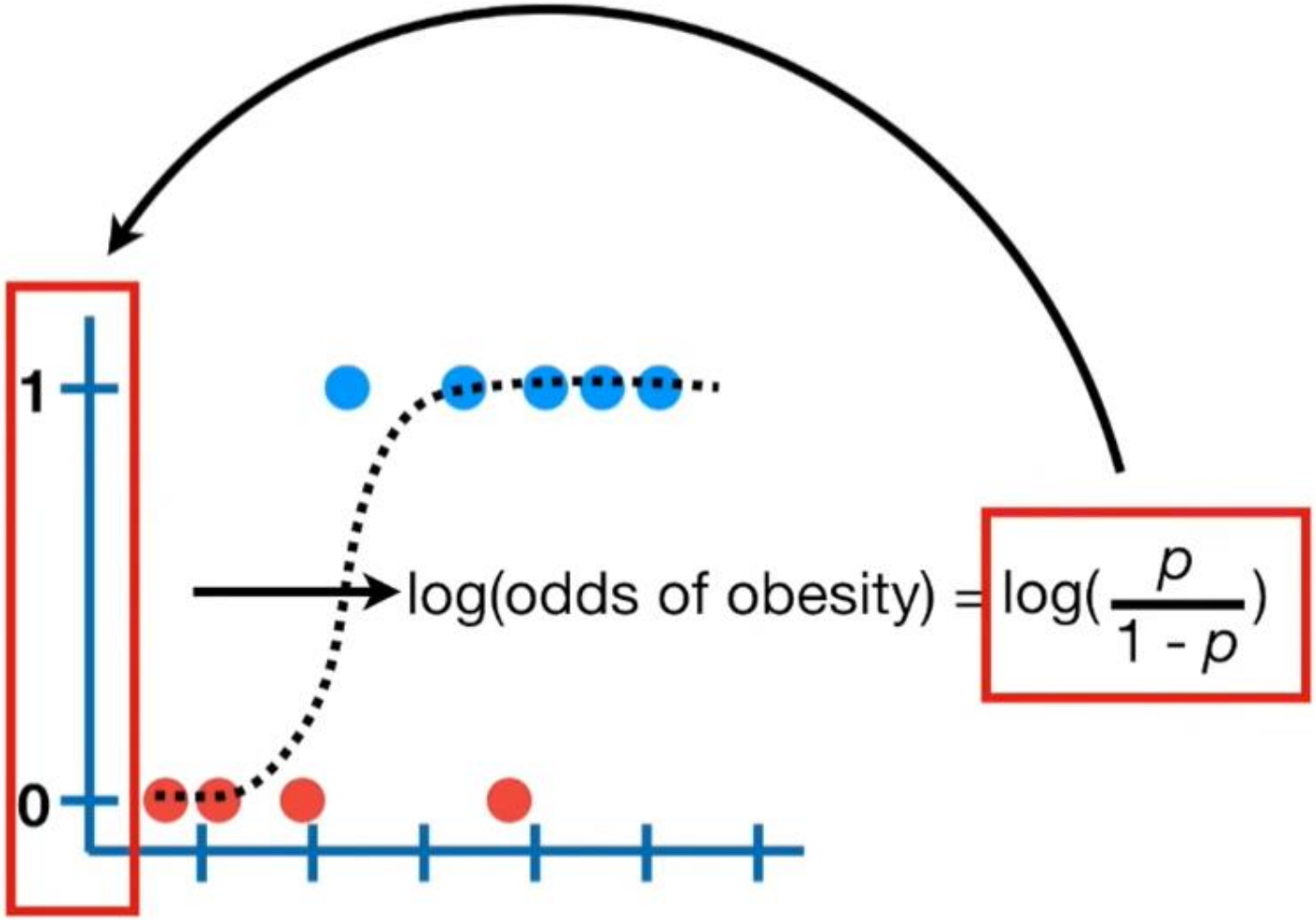
then ship the line and calculate the likelihood, again, and



Finally the curve with the maximum value for the likelihood is selected bam

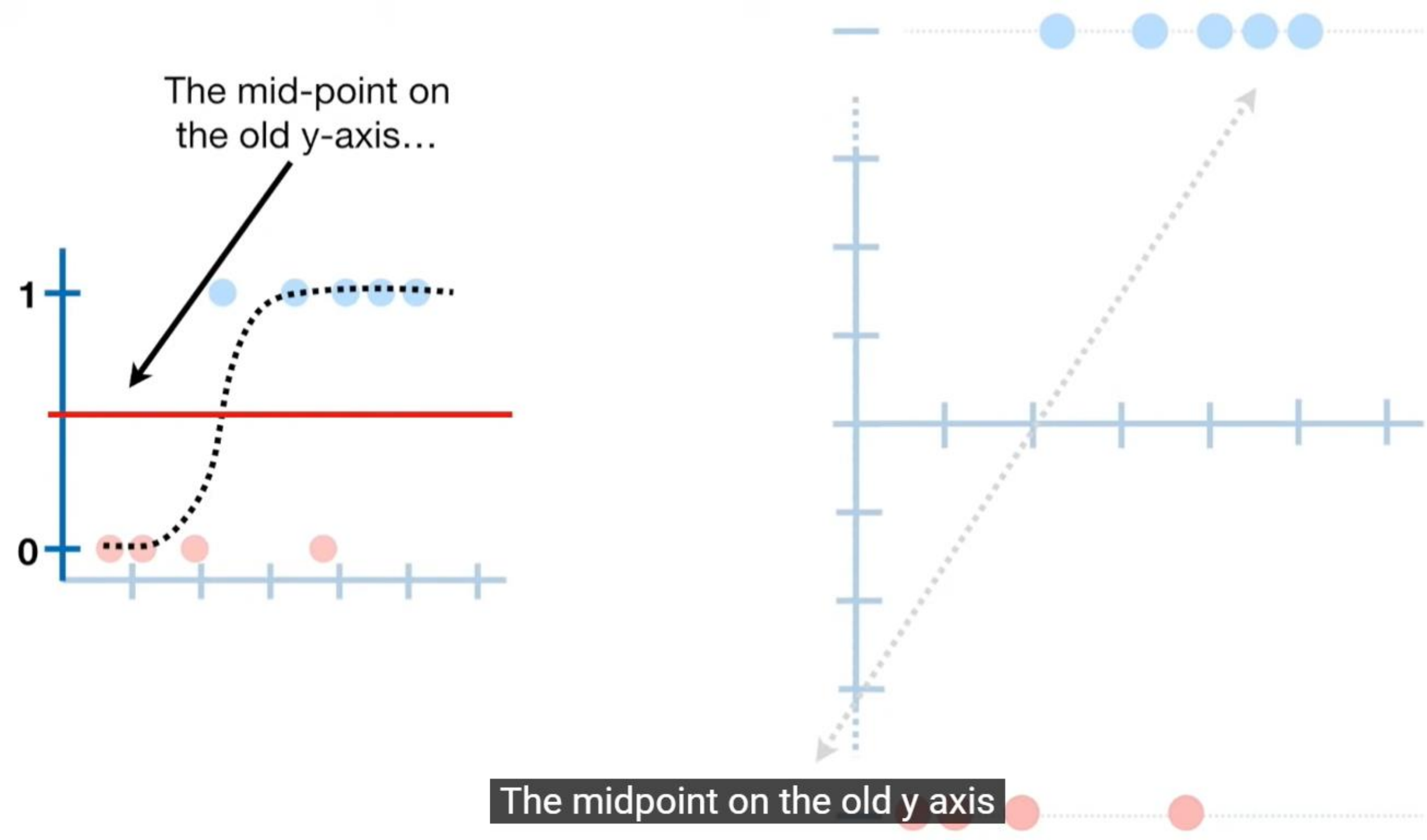


Now let's transform the y-axis from a probability of obesity scale to a log odds of obesity scale

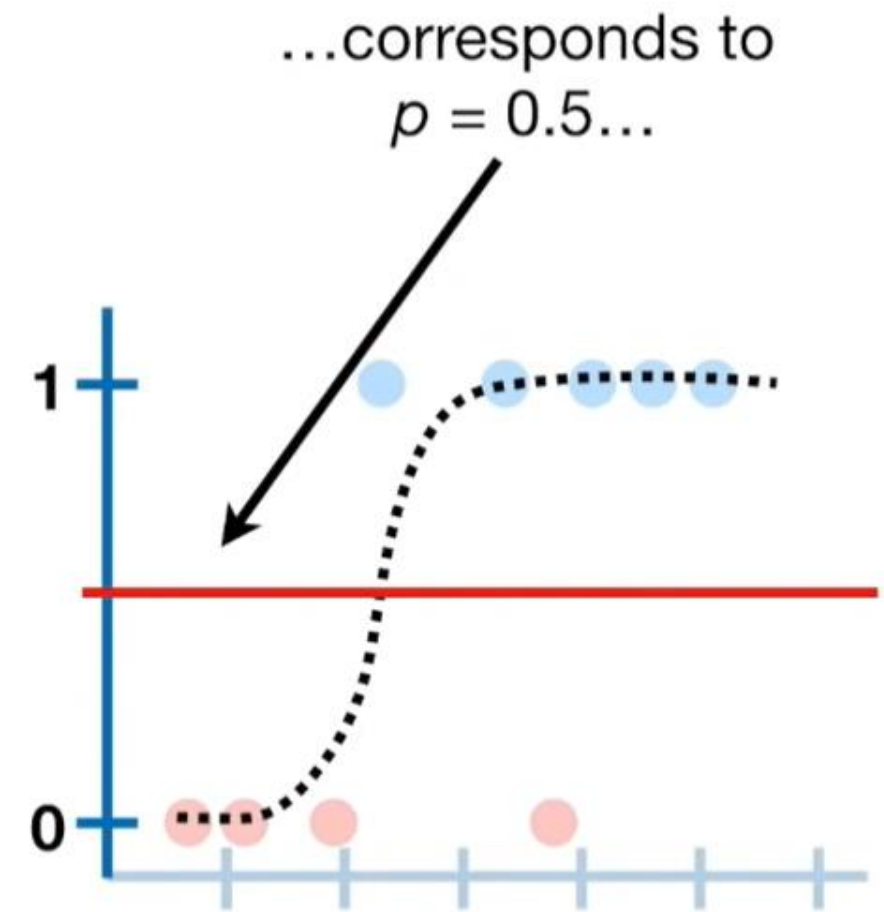


p, in this case, is the probability of a mouse being obese, and corresponds a value on the old y-axis between 0 and 1.

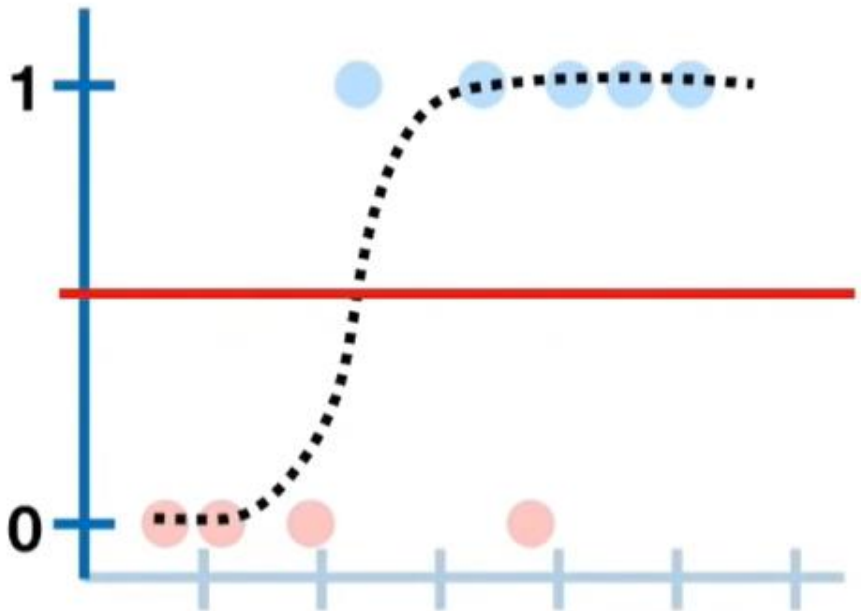
P in this case is the probability of a mouse being obese and



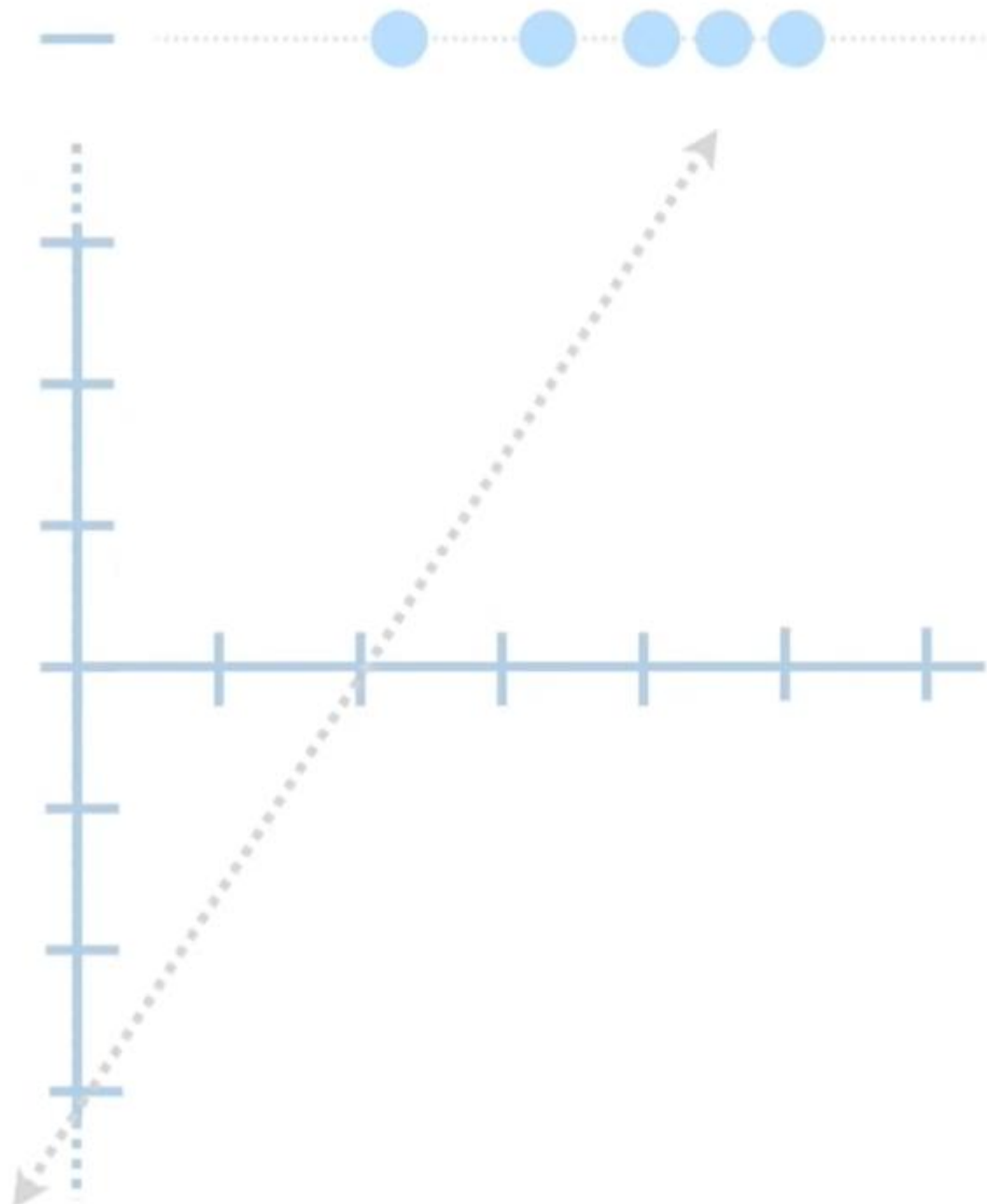
The midpoint on the old y axis


…corresponds to
p = 0.5…
1
0

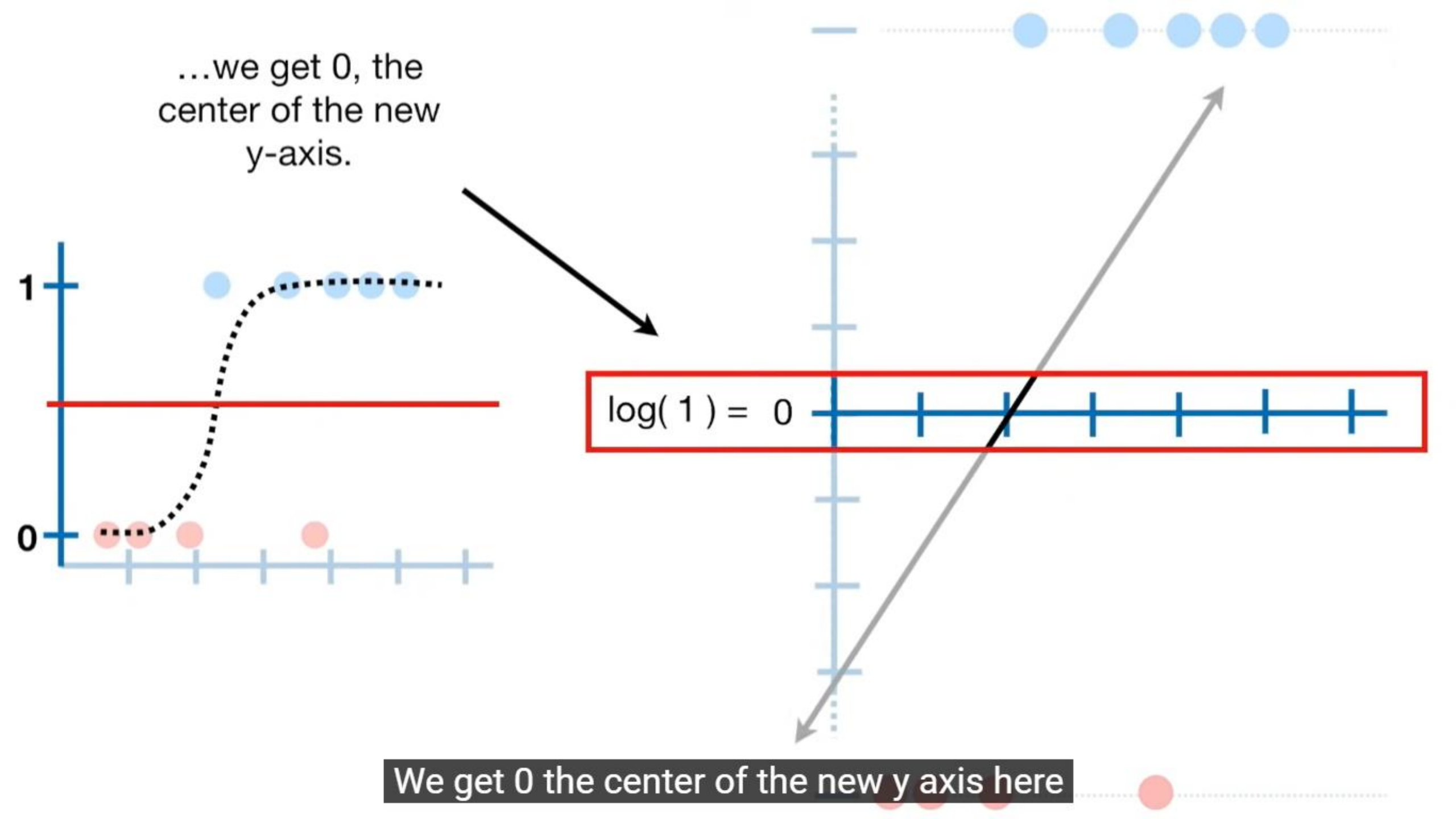
…and when we plug $p = 0.5$ into the logit formula and do the math…



$$\log(\frac{0.5}{1-0.5})$$

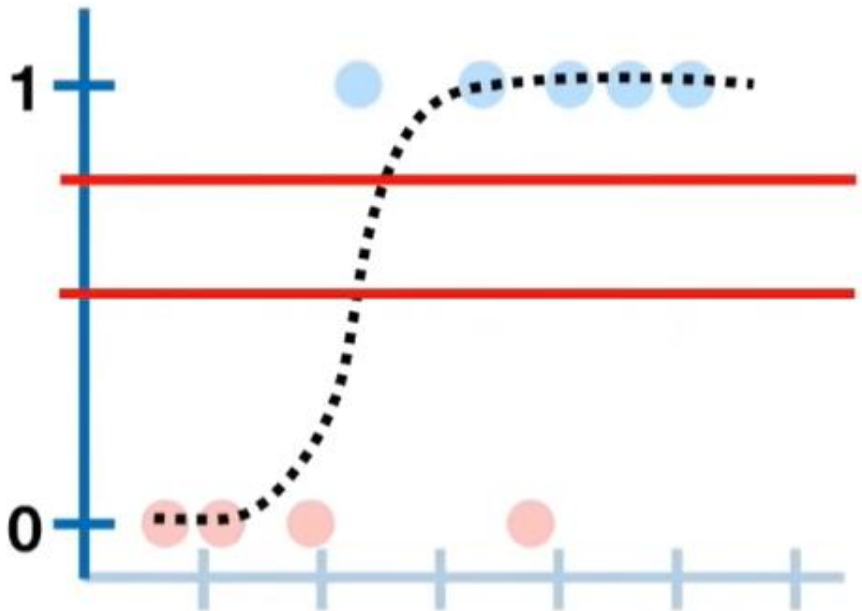


When we plug P equals 0.5 into the logit formula and do the math

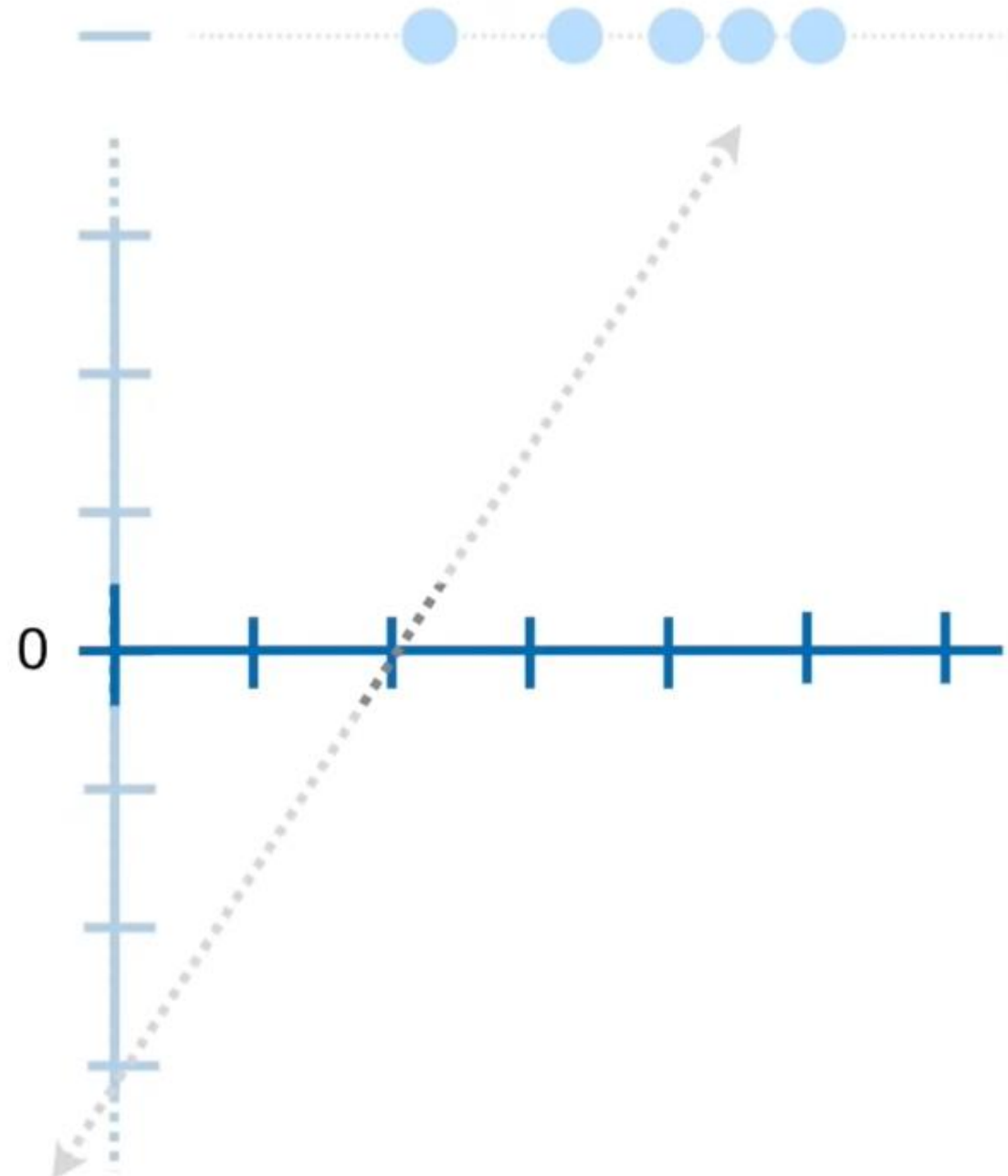


We get 0 the center of the new y axis here

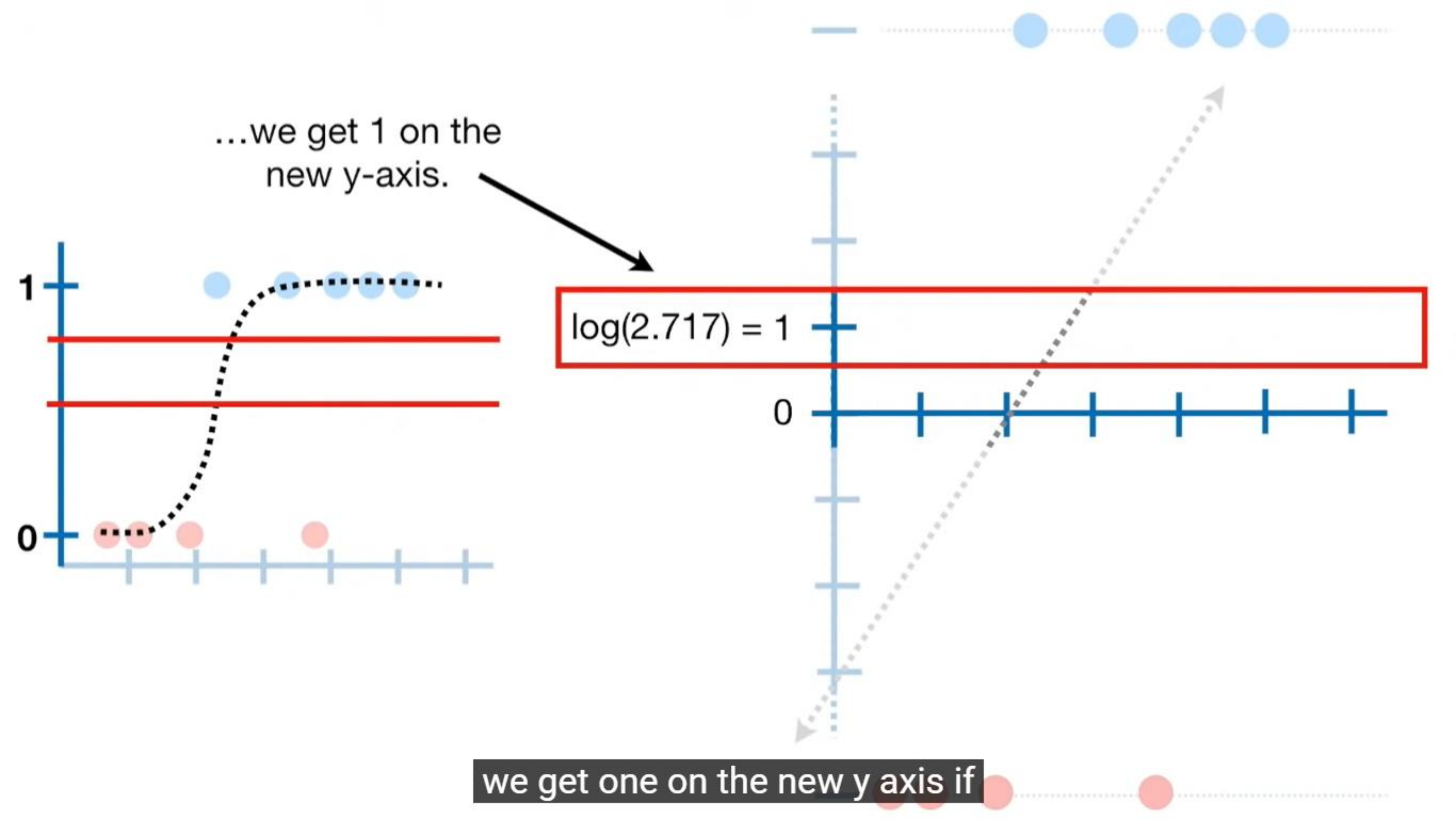
If we plug $p = 0.731$ into the logit function and do the math…



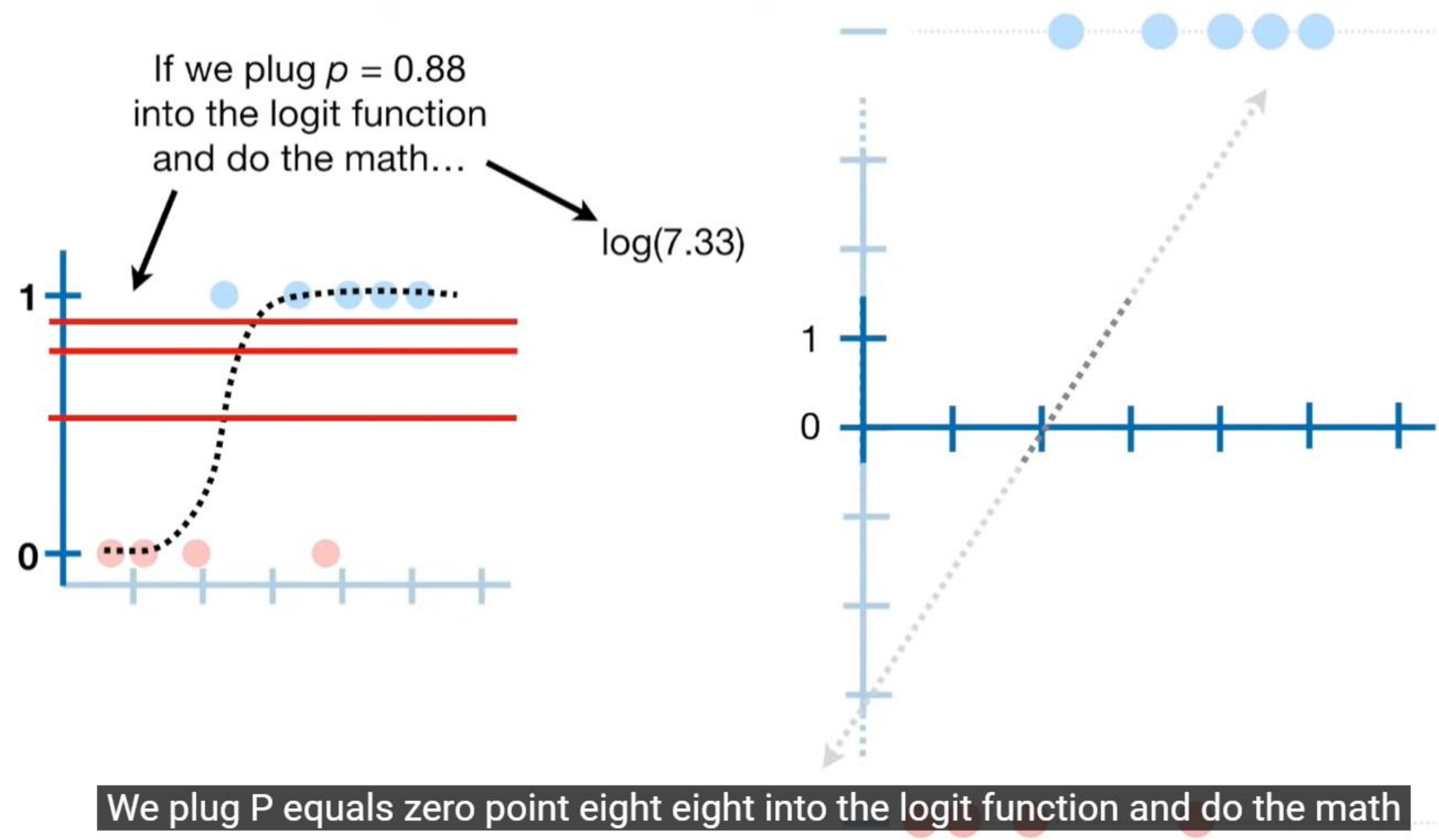
log(2.717)



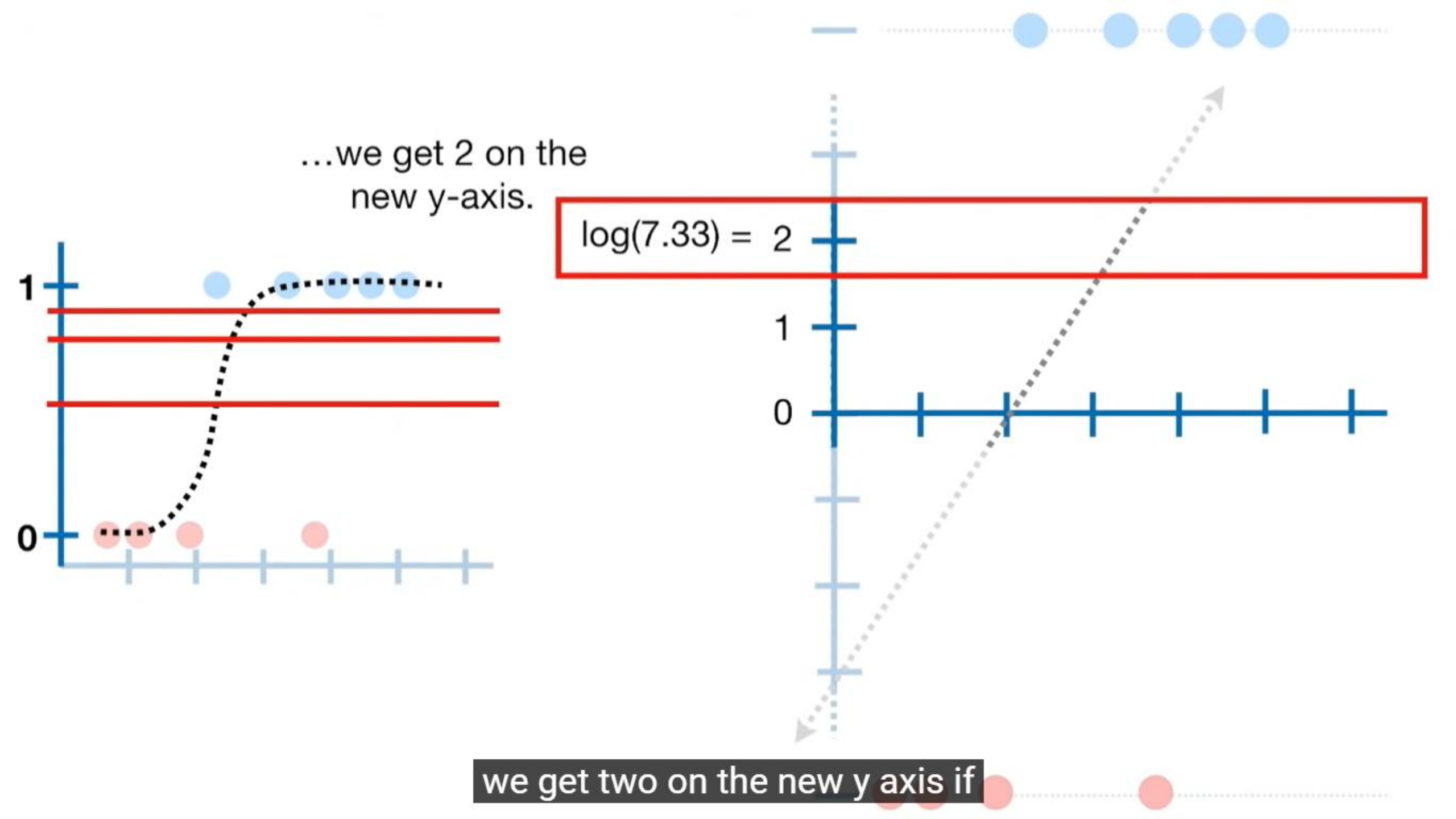
We plug P equals zero point seven three one into the logit function and do the math



we get one on the new y axis if

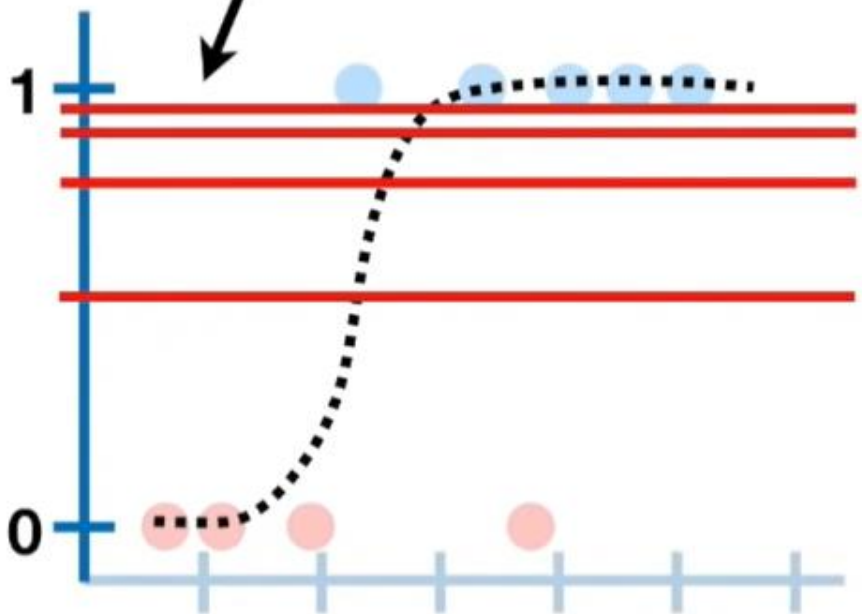


We plug P equals zero point eight eight into the logit function and do the math

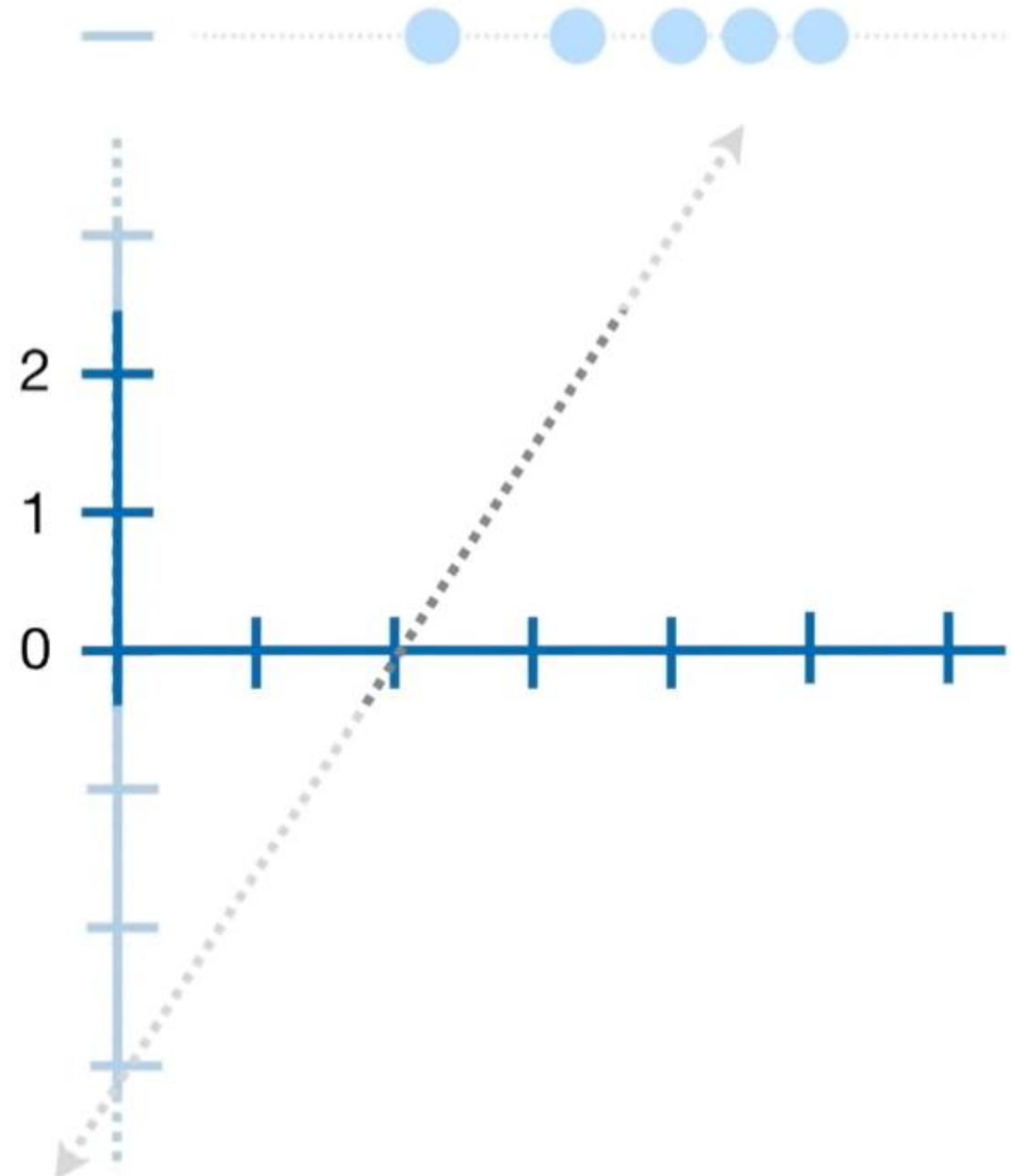


we get two on the new y axis if

If we plug $p = 0.95$ into the logit function and do the math…

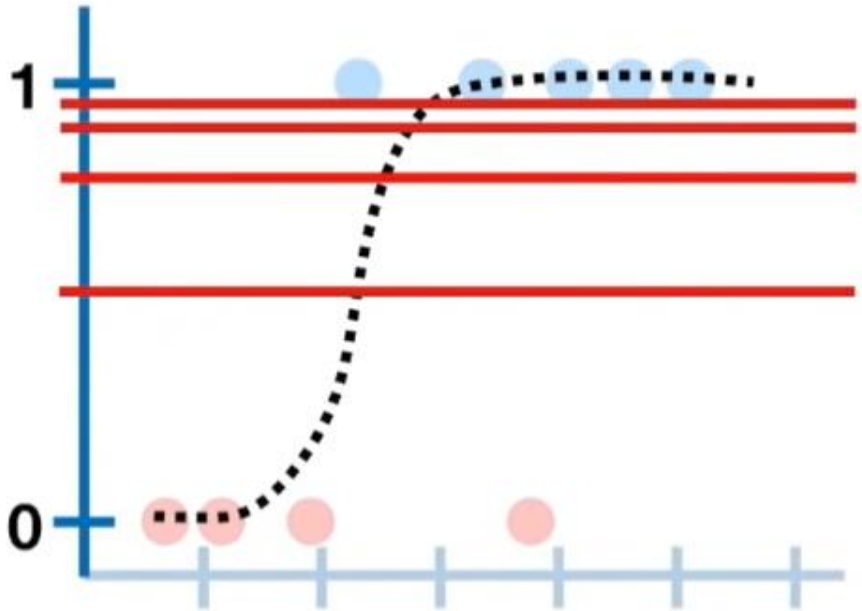


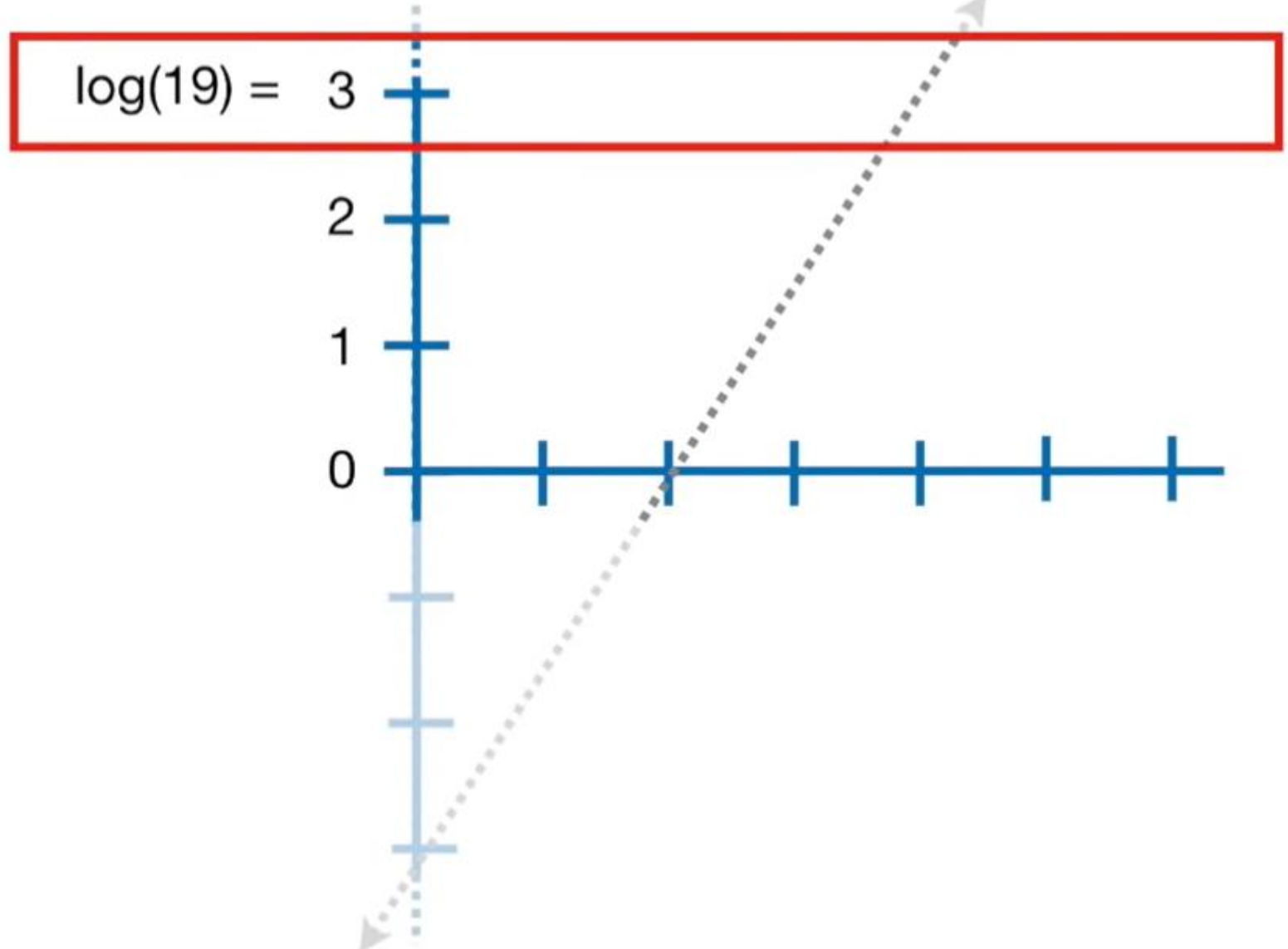
log(19)



We plug P equals zero point nine five into the logit function and do the math

…we get 3 on the new y-axis.

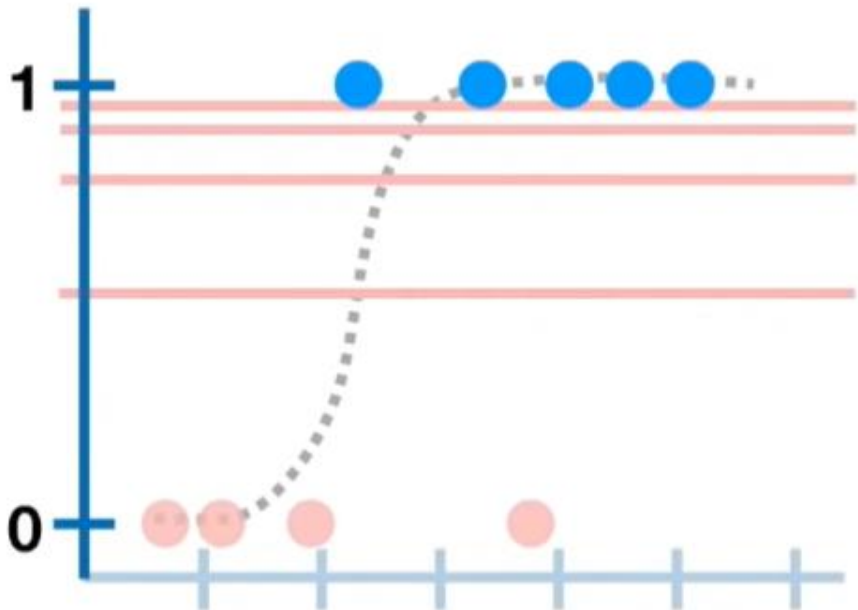


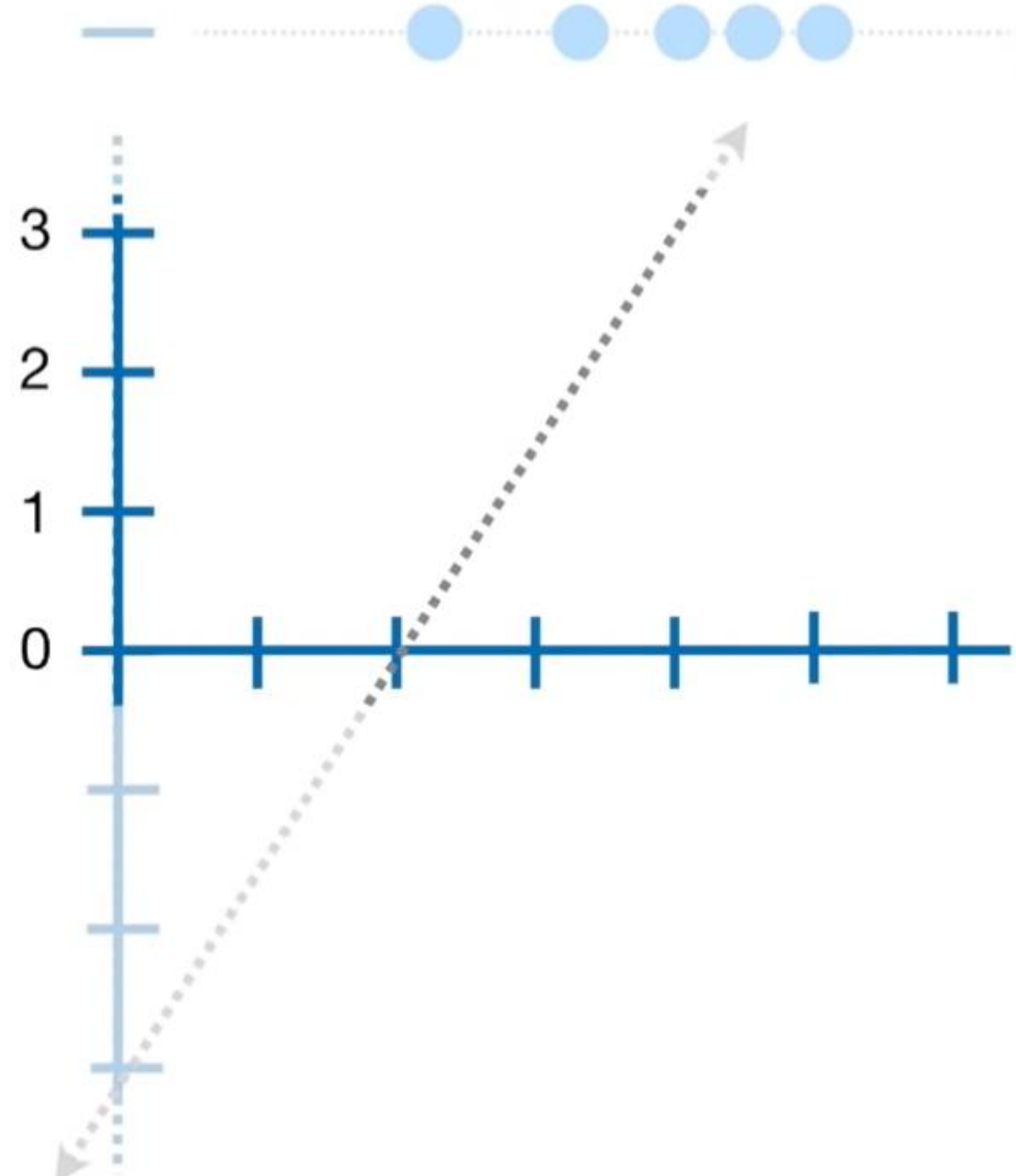


We get three on the new y axis

If we plug $p = 1$ into the logit function and do the math…

$$\log\left(\frac{1}{1 - 1}\right)$$





If we plug P equals one into the logit function and do the math

…technically, you can't divide by 0, however… → $\log(\frac{1}{0})$

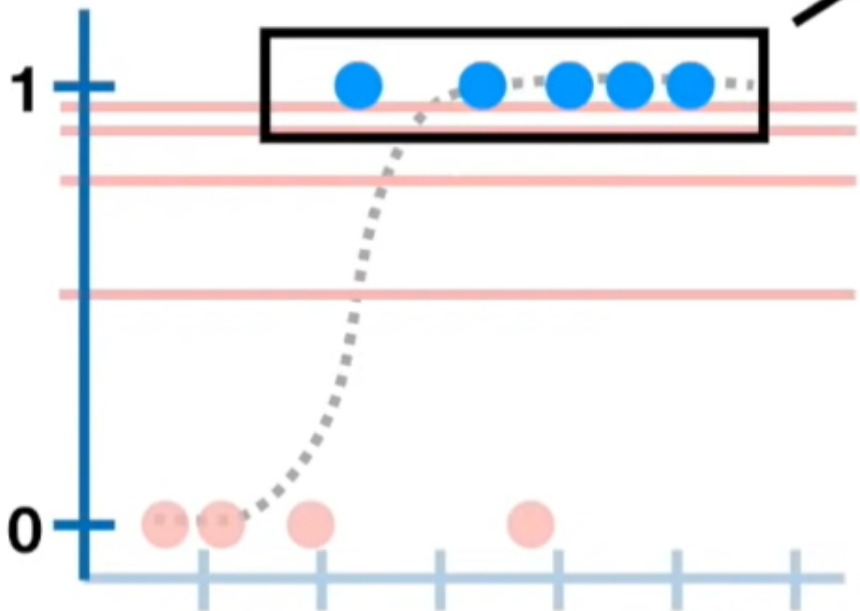
=

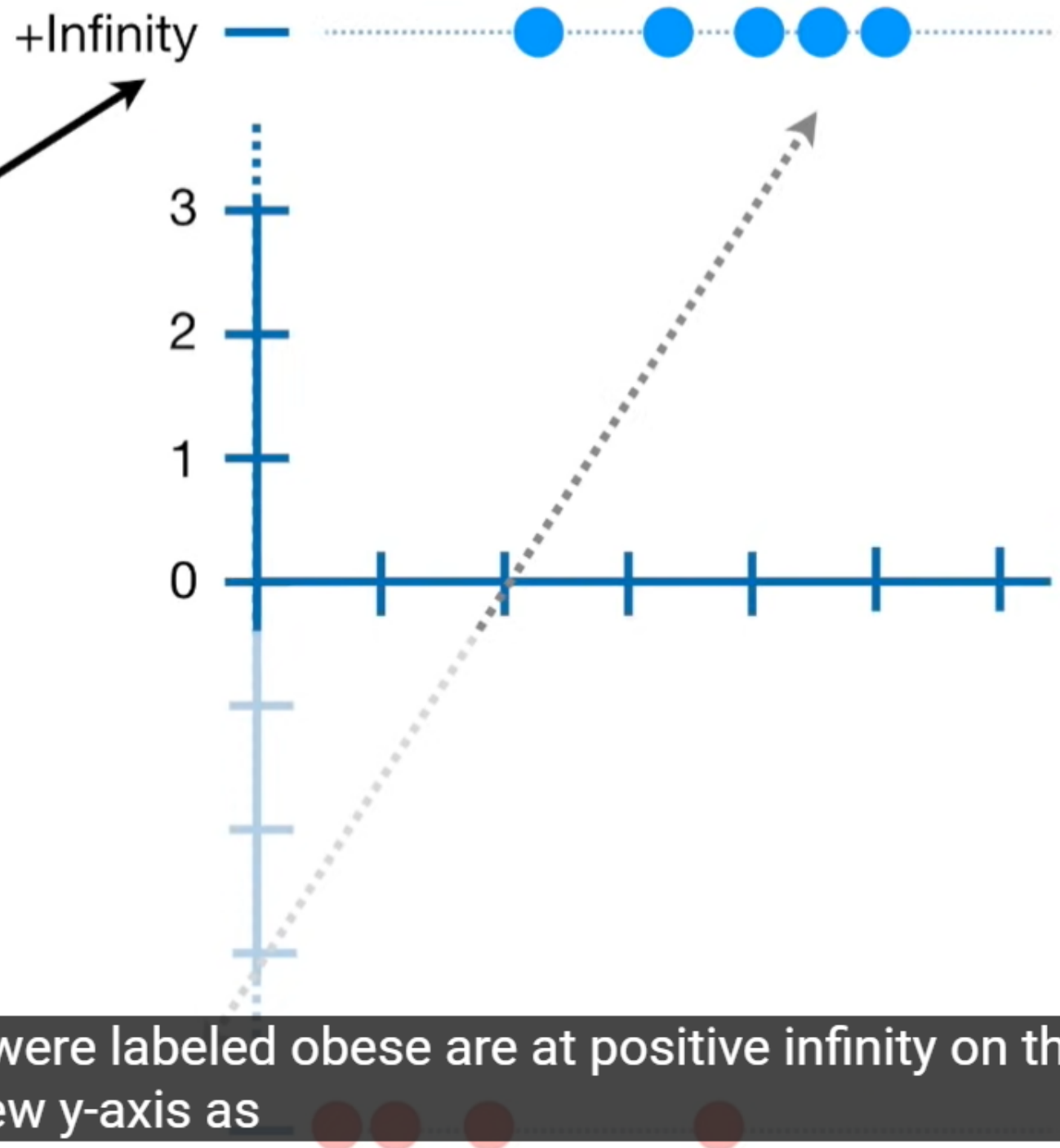
$\log(1) - \log(0)$

…and since
"something - negative infinity = positive infinity",
this whole thing is
equal to positive infinity.

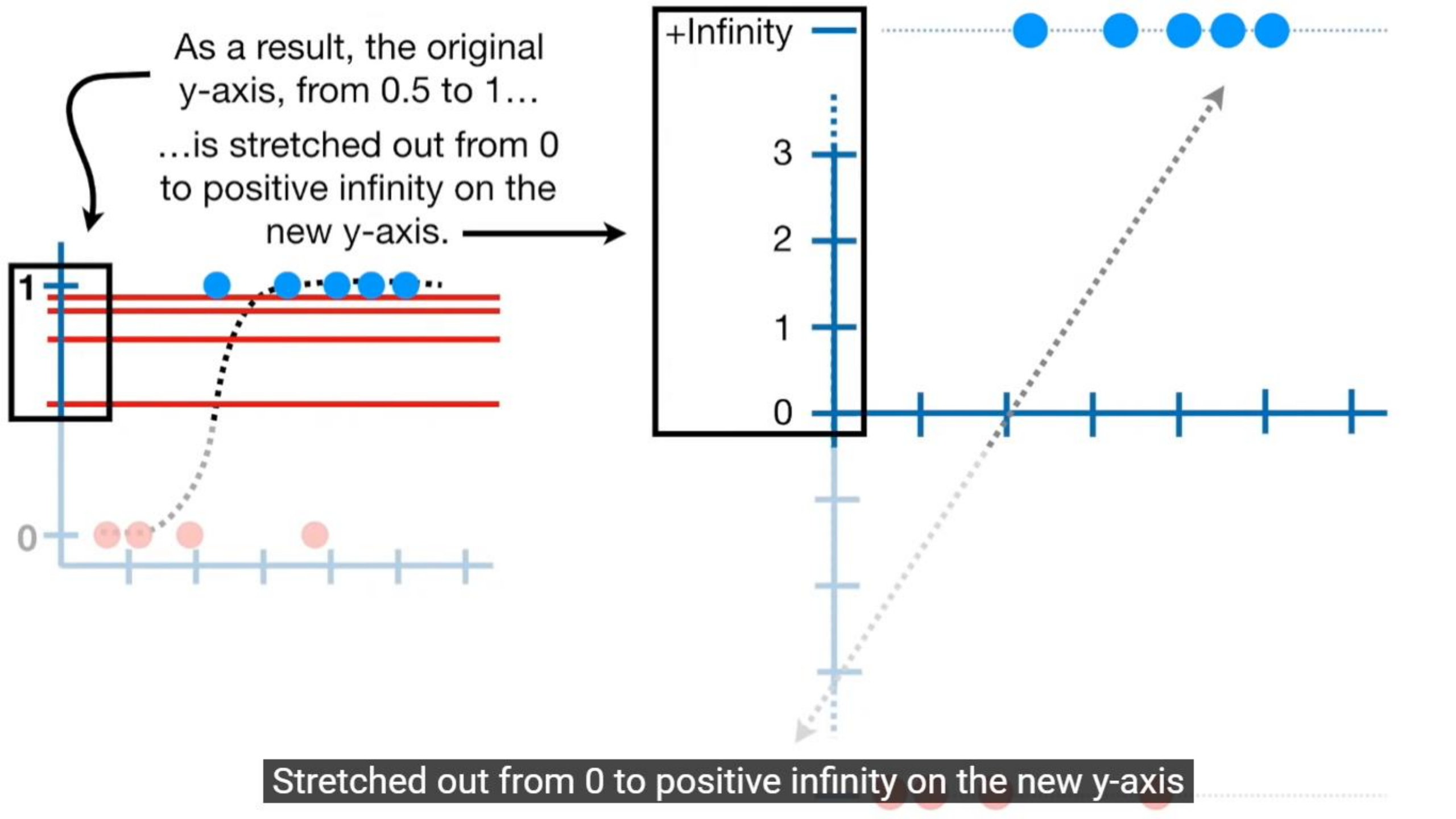
Since something minus negative infinity equals positive infinity this whole thing is equal to positive infinity

This means the original samples that were labeled “obese” are at positive infinity on the new y-axis.

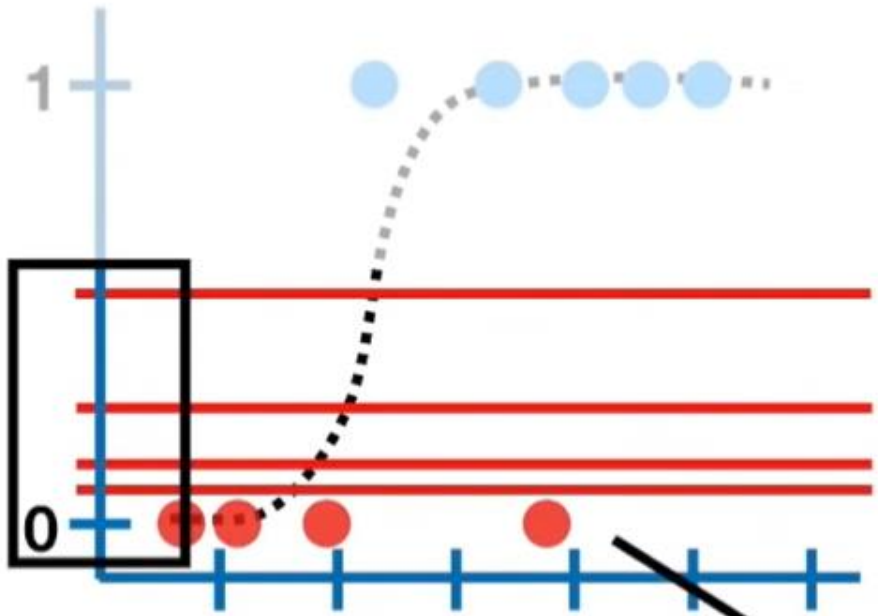


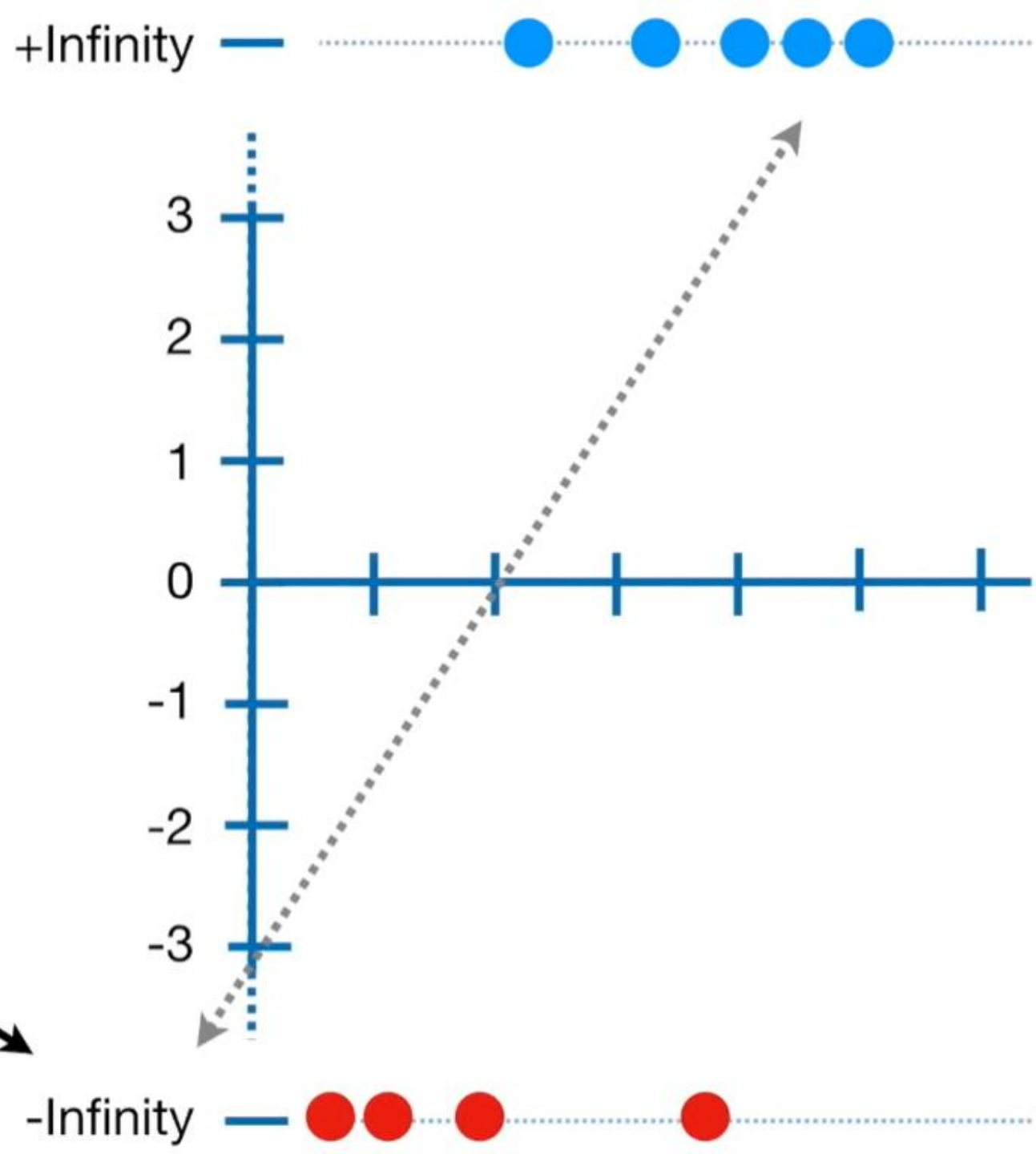


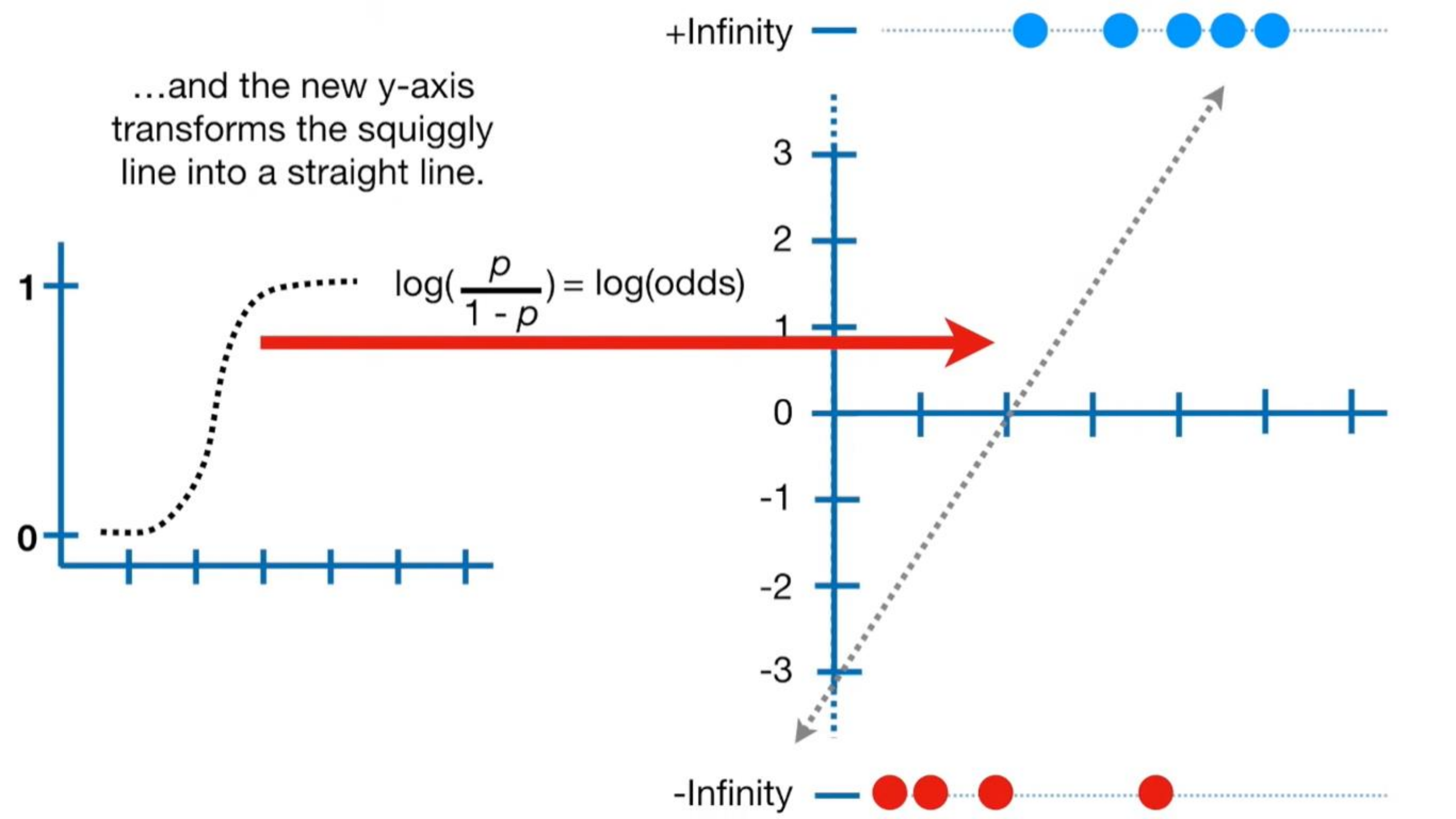
This means the original samples that were labeled obese are at positive infinity on the new y-axis as


As a result, the original y-axis, from 0.5 to 1…
…is stretched out from 0 to positive infinity on the new y-axis.
1
0
+Infinity
3
2
1
0
Stretched out from 0 to positive infinity on the new y-axis

Similarly, 0.5 to 0 on the old y-axis is stretched out from 0 to -Infinity on the new y-axis.

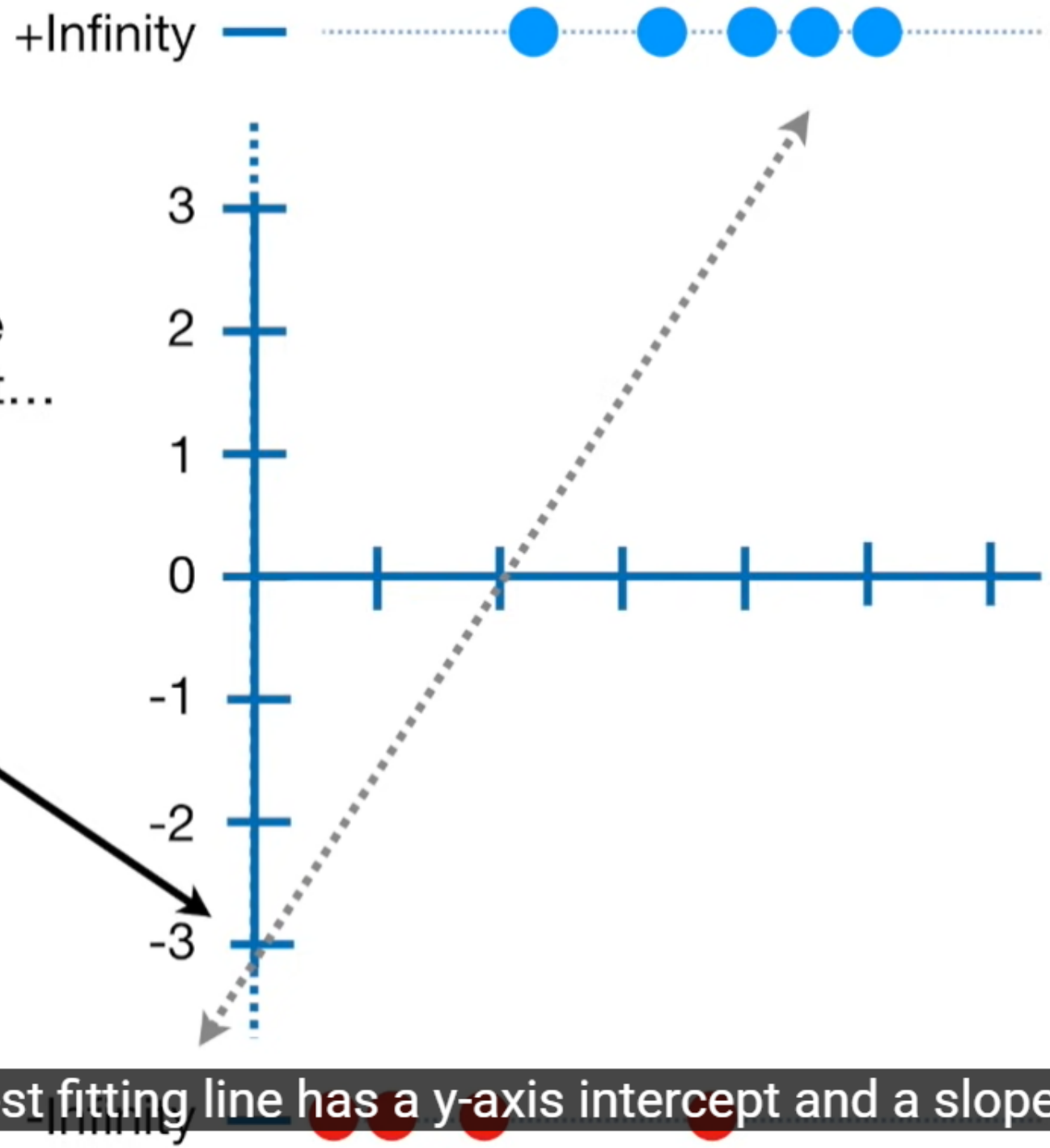





+Infinity
…and the new y-axis transforms the squiggly line into a straight line.
$\log(\frac{p}{1-p}) = \log(odds)$
1
0
3
2
1
0
-1
-2
-3
-Infinity

$y = \mathbf{-3.48} + \mathbf{1.83} \times \text{weight}$

Just like with linear regression, the best fitting line has a y-axis intercept…

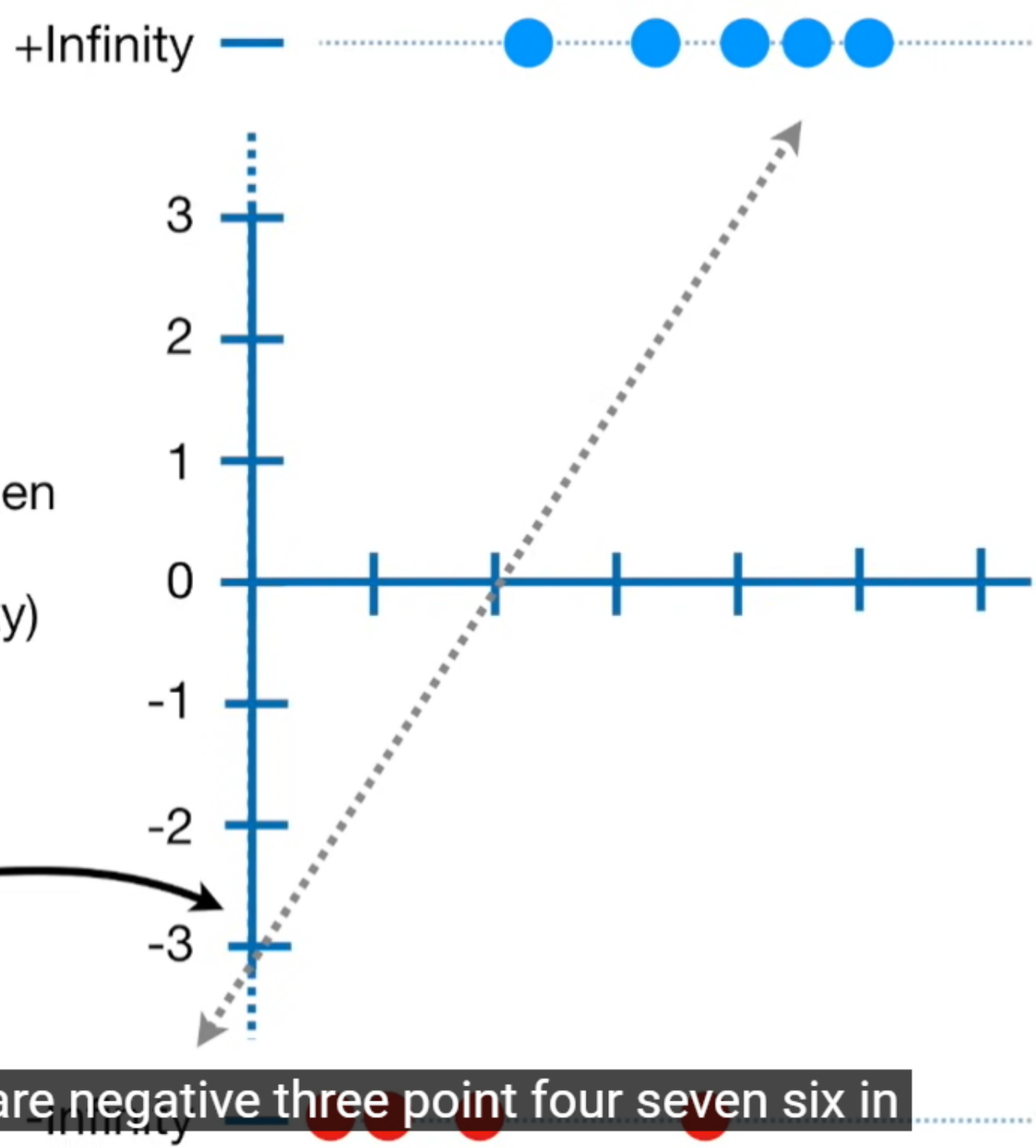


Just like with linear regression the best fitting line has a y-axis intercept and a slope

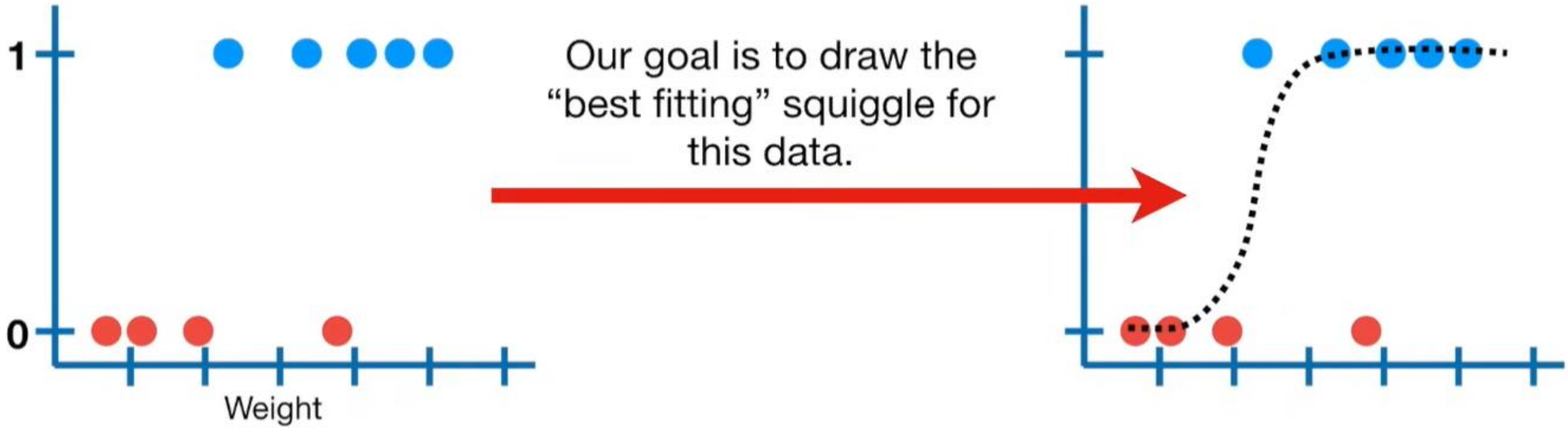
$y = \mathbf{-3.48} + \mathbf{1.83} \times \text{weight}$

```
Coefficients:
             Estimate
(Intercept)    -3.476
```

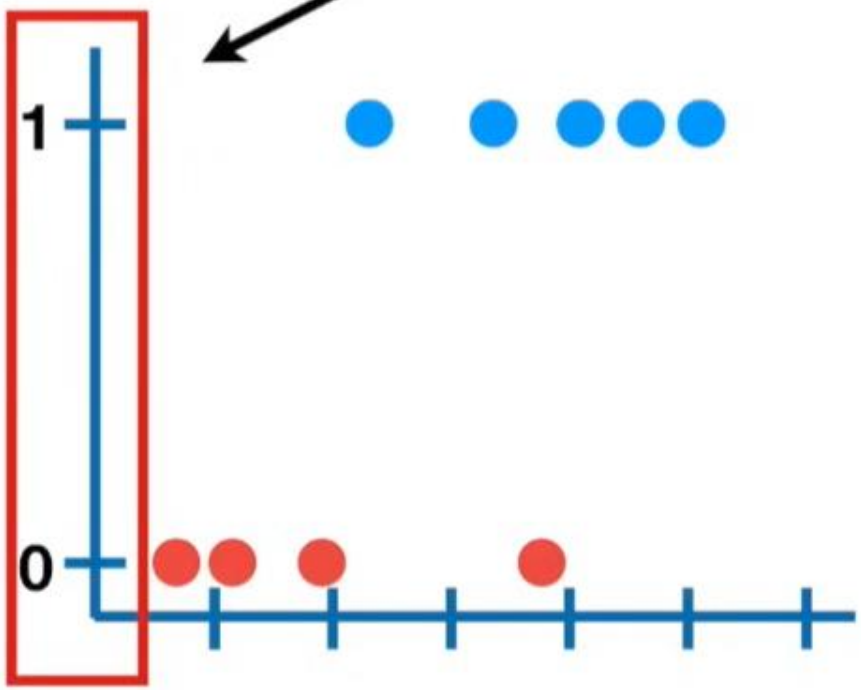
…it means that when weight = 0, the log(odds of obesity) are -3.476.

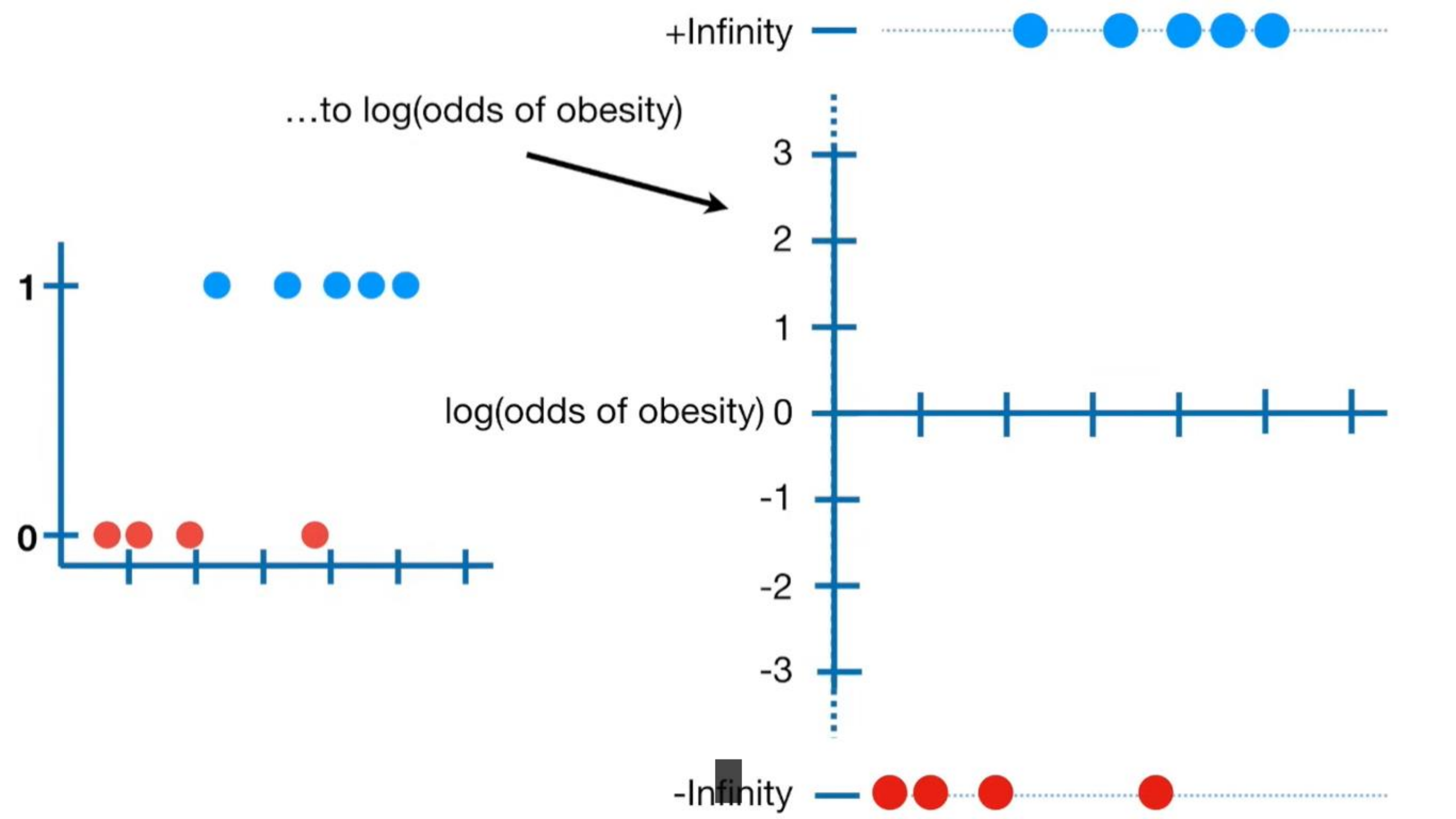


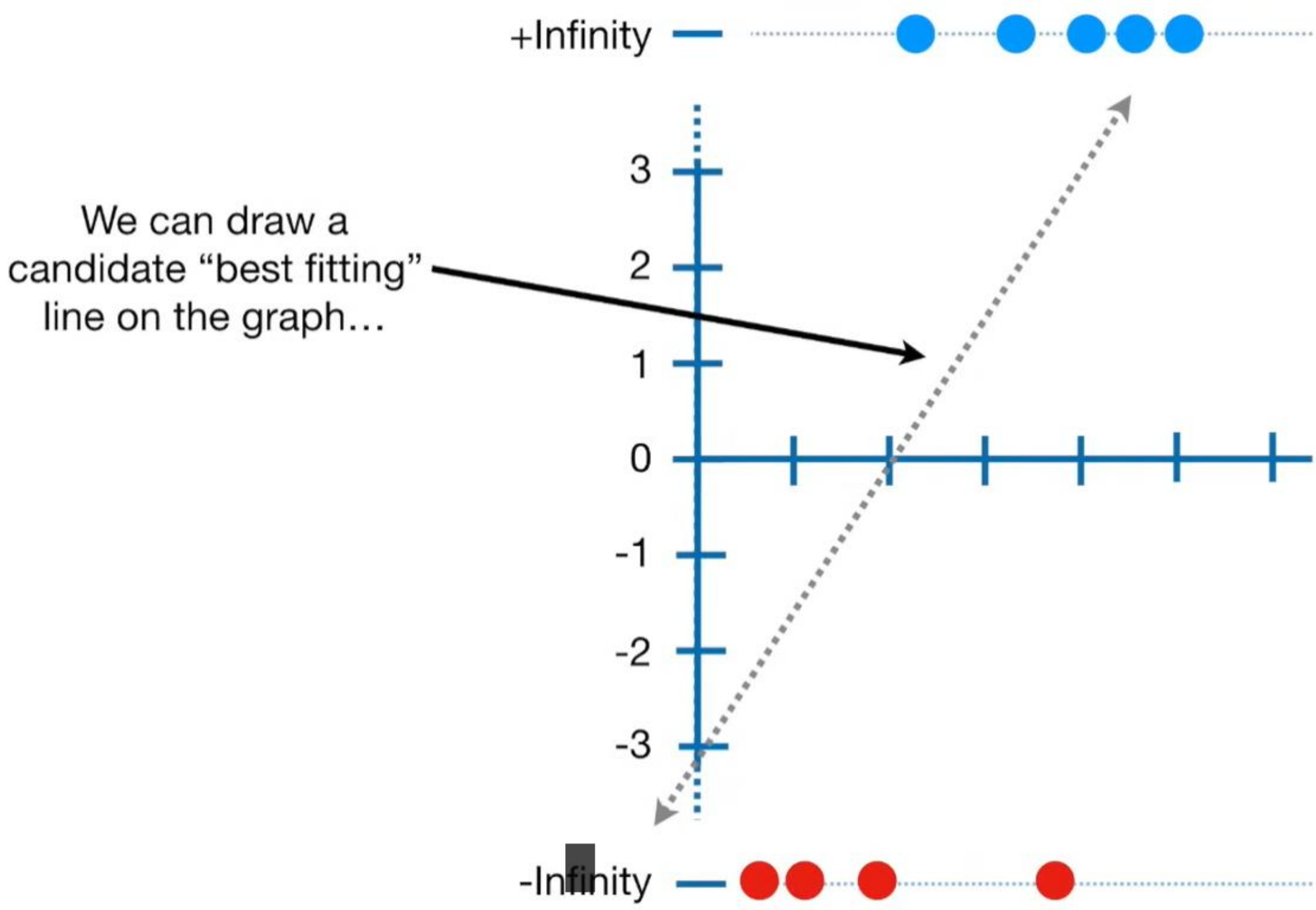
the log of the odds of obesity are negative three point four seven six in


1
0
Weight
Our goal is to draw the
“best fitting” squiggle for
this data.

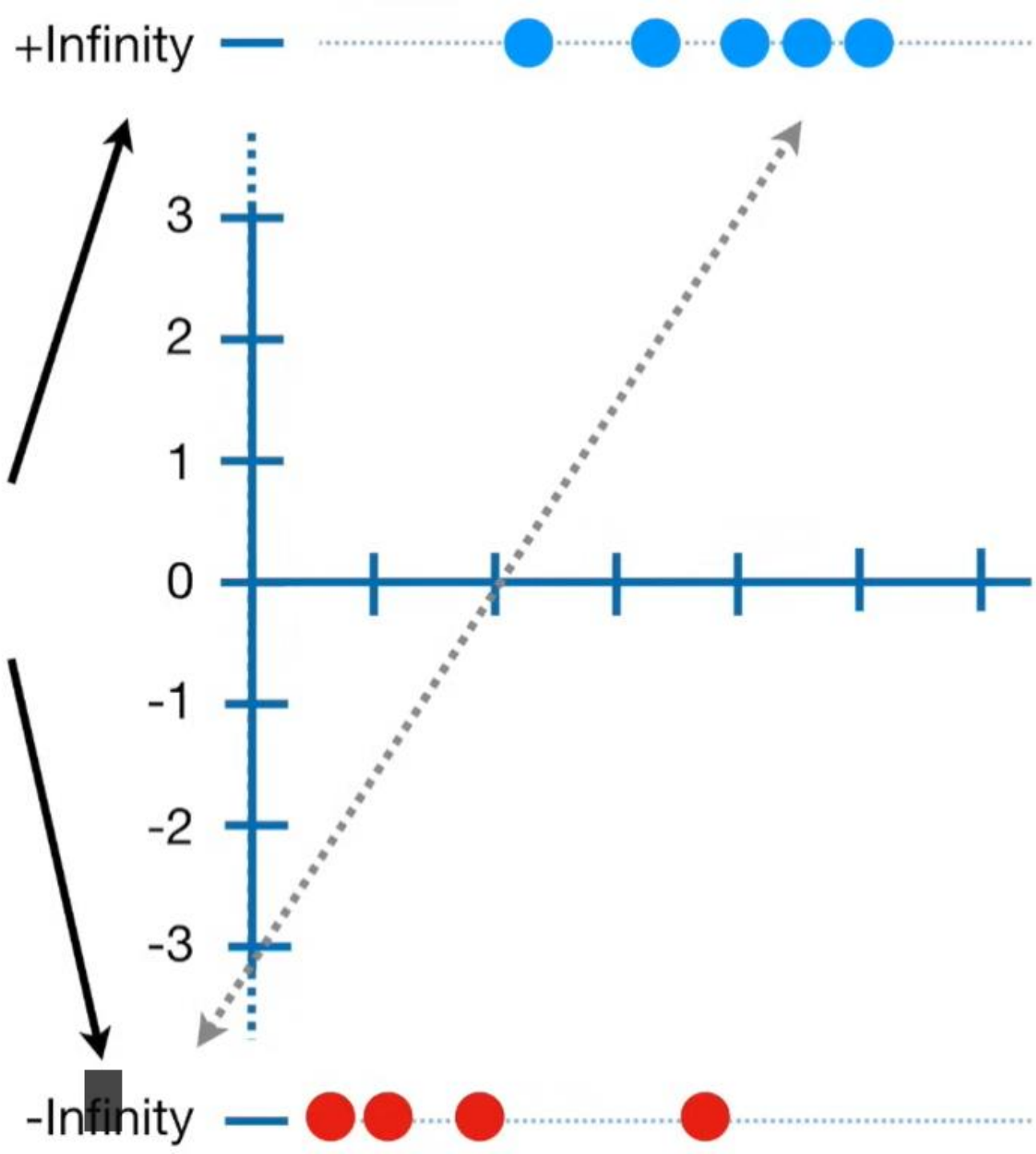
As we know, in logistic regression, we transform the y-axis from the probability of obesity…



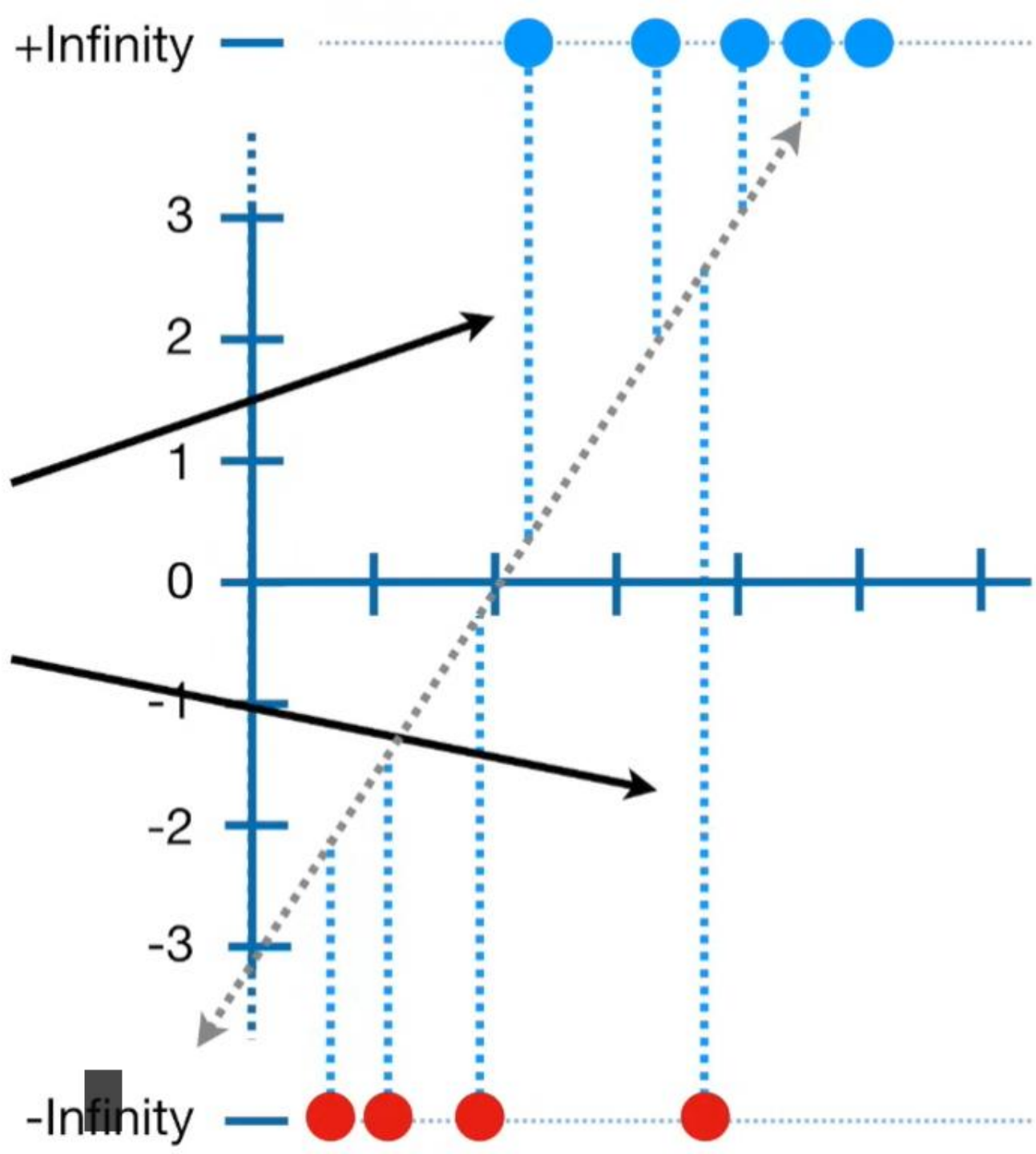

+Infinity
…to log(odds of obesity)
3
2
1
log(odds of obesity) 0
-1
-2
-3
1
0
-Infinity


+Infinity
We can draw a candidate “best fitting” line on the graph…
3
2
1
0
-1
-2
-3
-Infinity

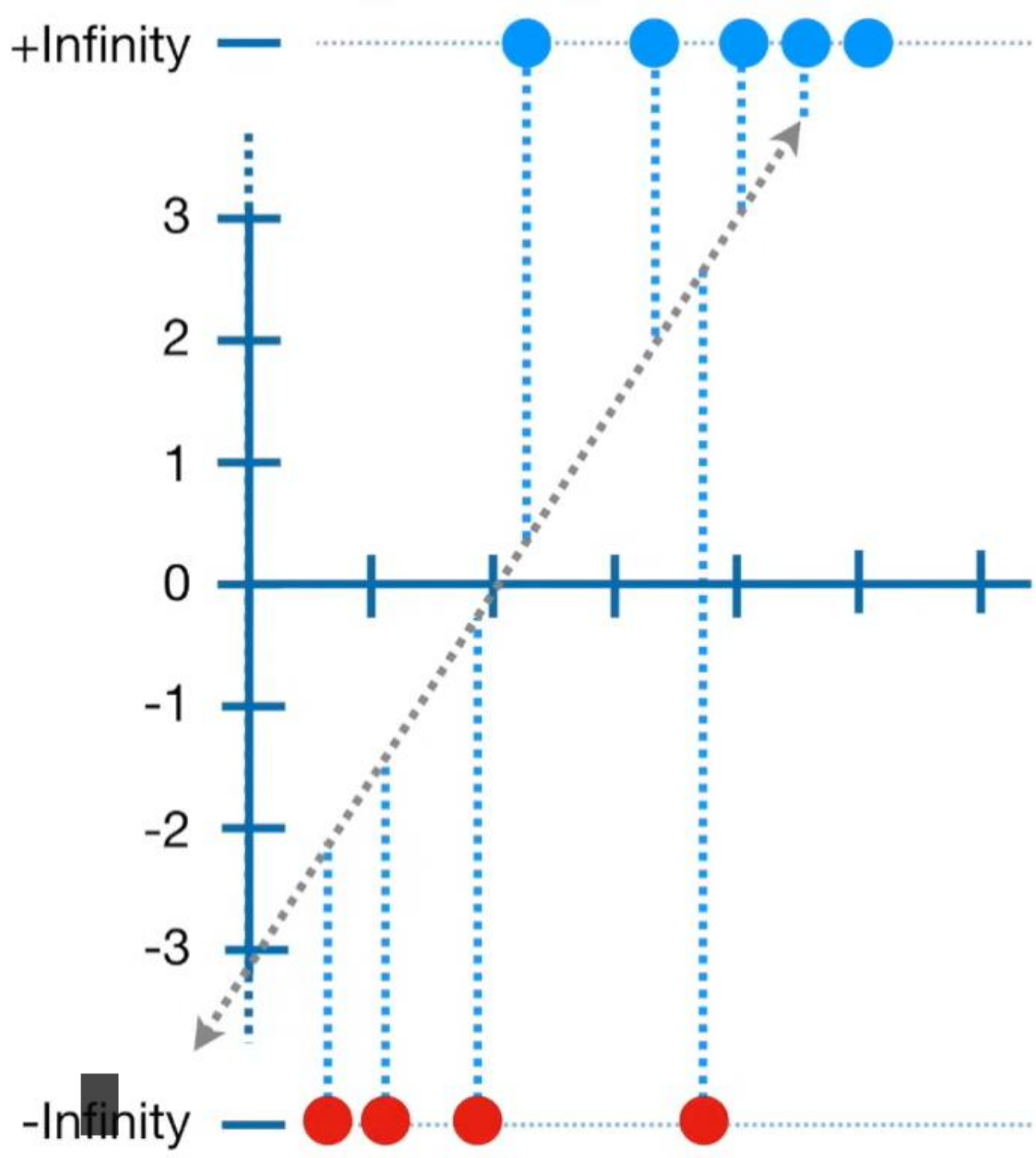
The only problem is that the transformation pushes the raw data to positive and negative infinity…



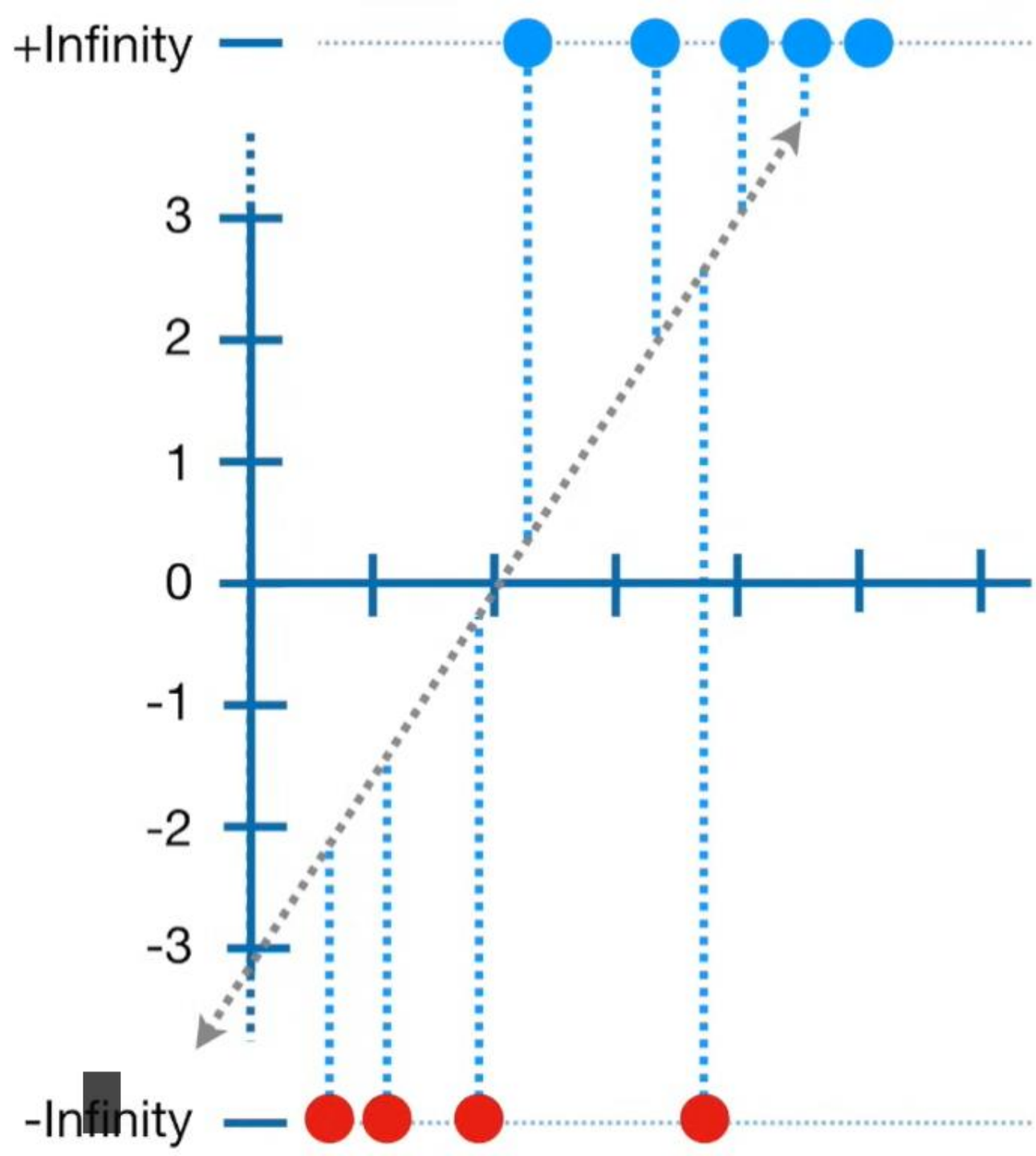
…and this means that the residuals (the distance from the data points to the line) are also equal to positive and negative infinity…



…and this means we can't use least-squares to find the best fitting line.

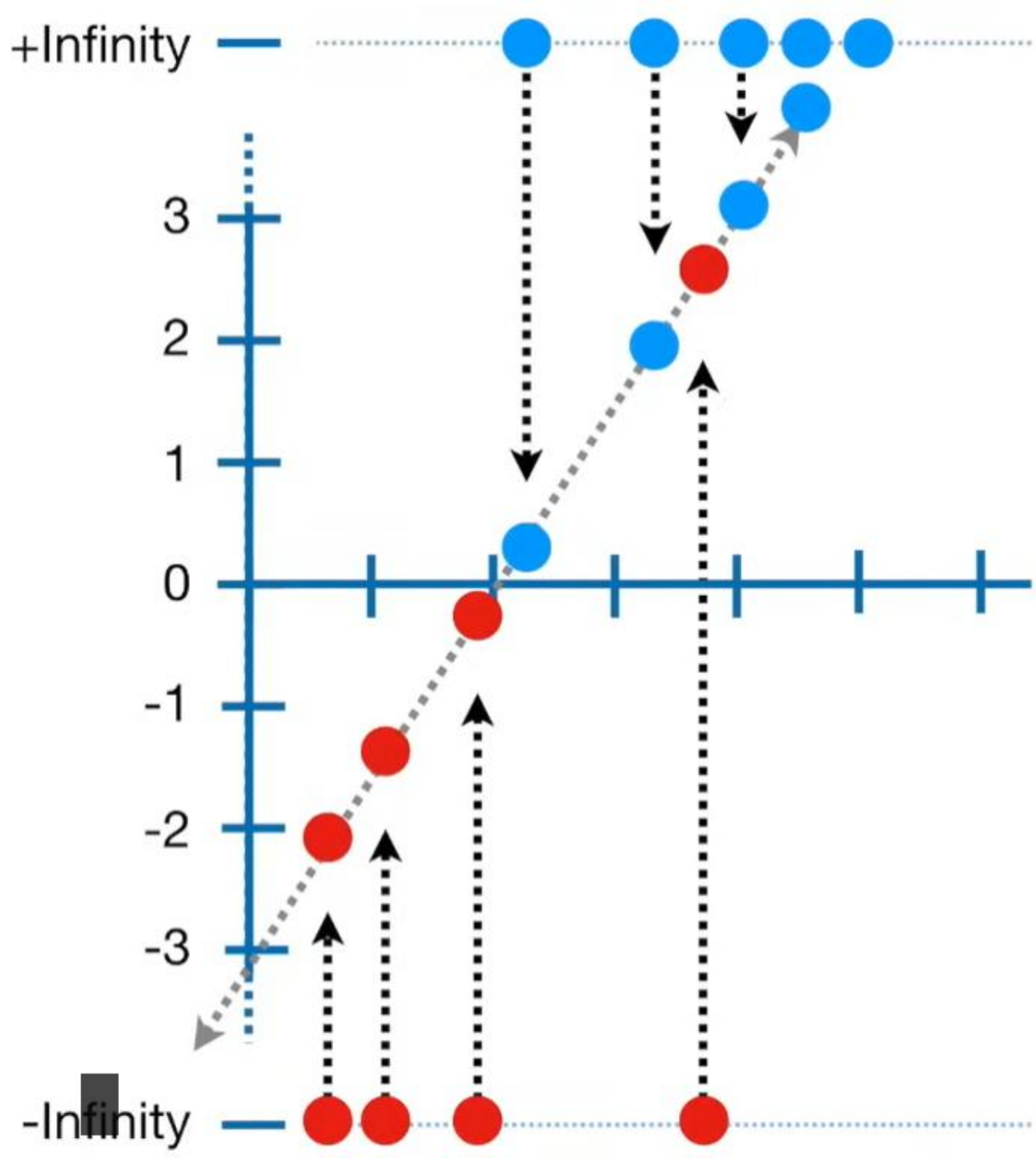


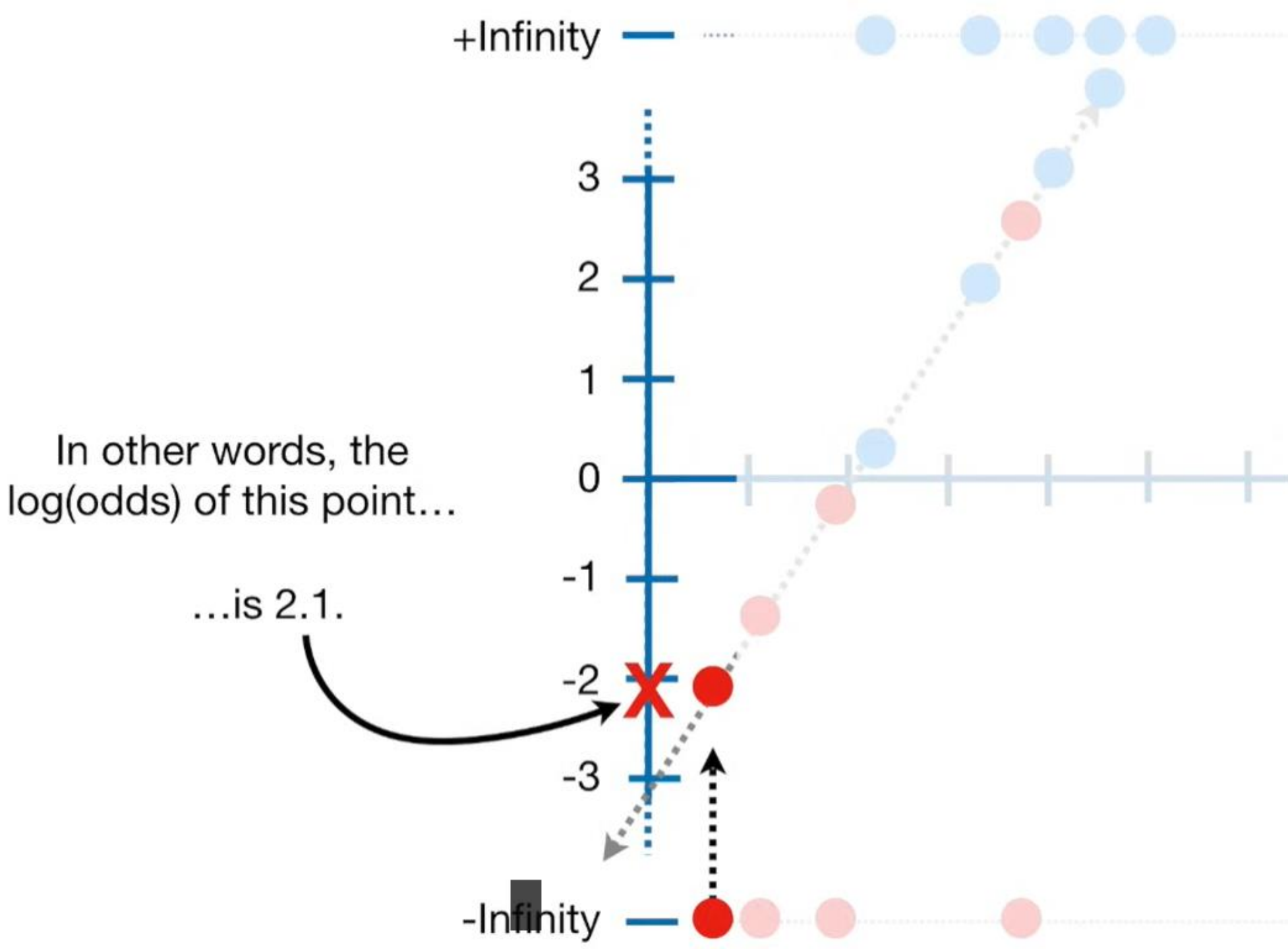
Instead, we use maximum likelihood…



The first thing we do is project the original data points onto the candidate line.

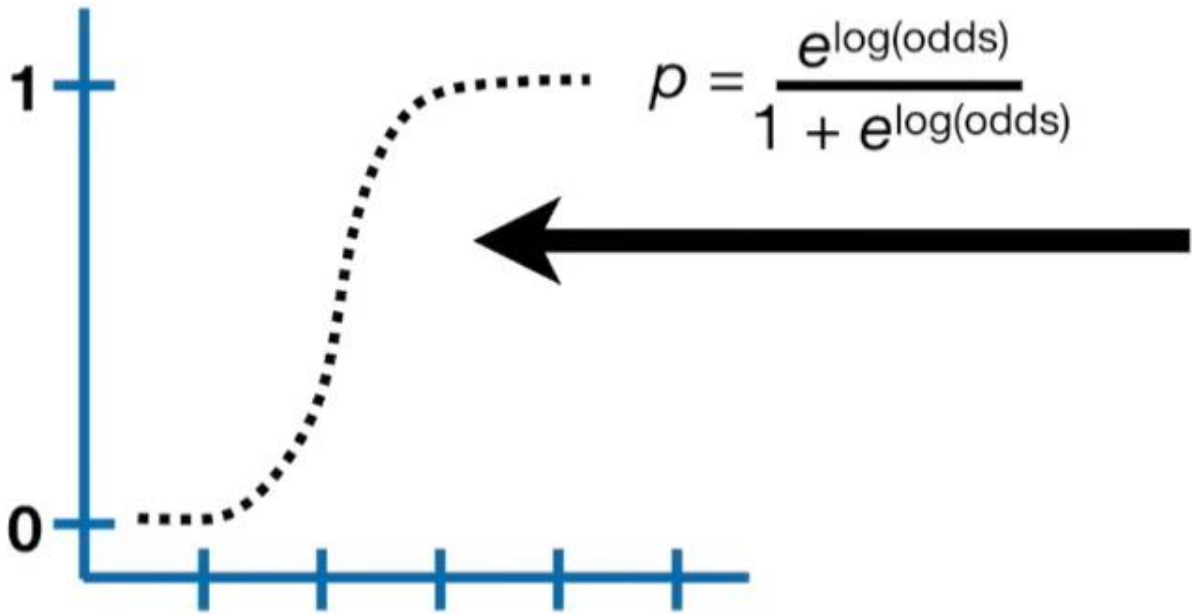
This gives each sample a candidate log(odds) value.

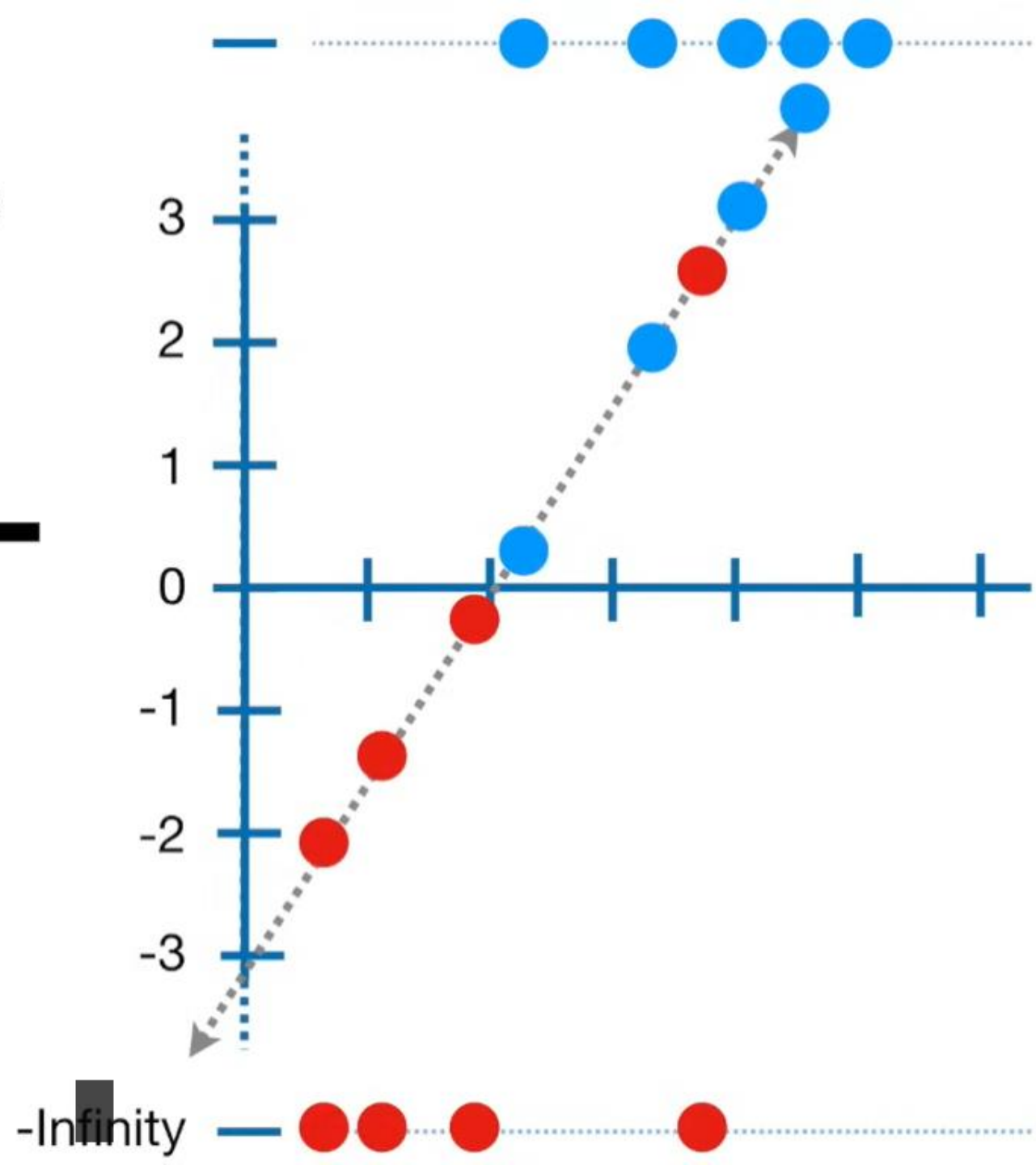



+Infinity
3
2
1
0
-1
-2
-3
-Infinity
In other words, the
log(odds) of this point…
…is 2.1.

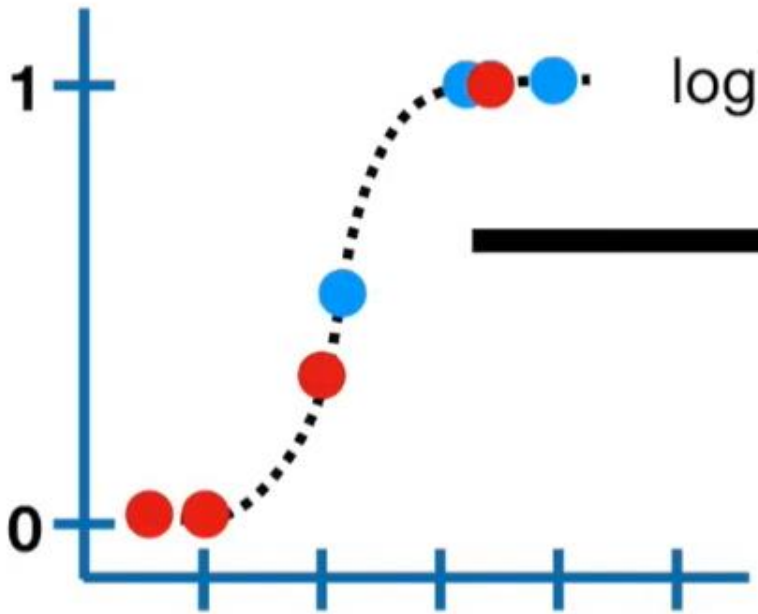
Then we transform the candidate log(odds) to candidate probabilities using this fancy looking formula…

$$p = \frac{e^{\log(odds)}}{1 + e^{\log(odds)}}$$

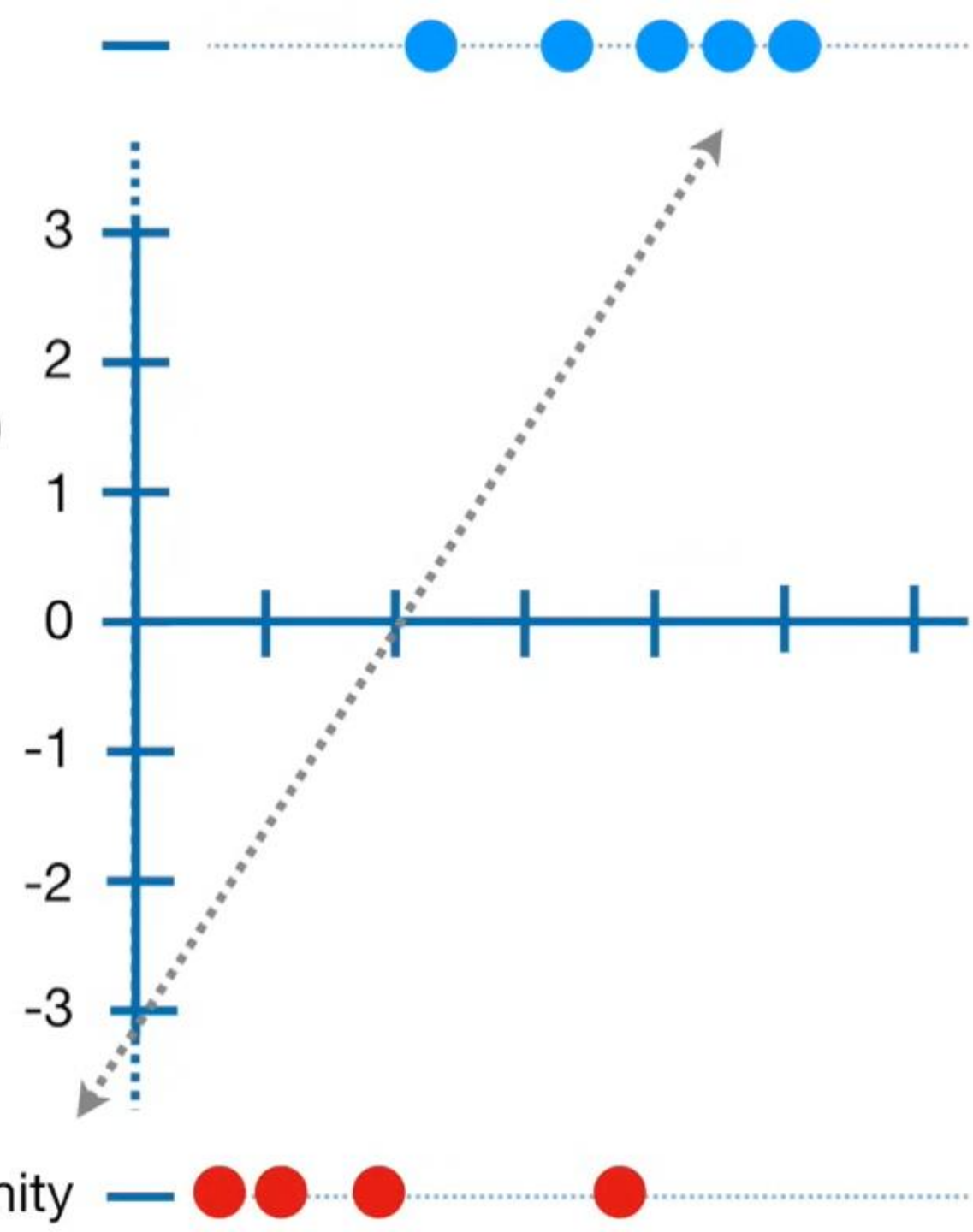


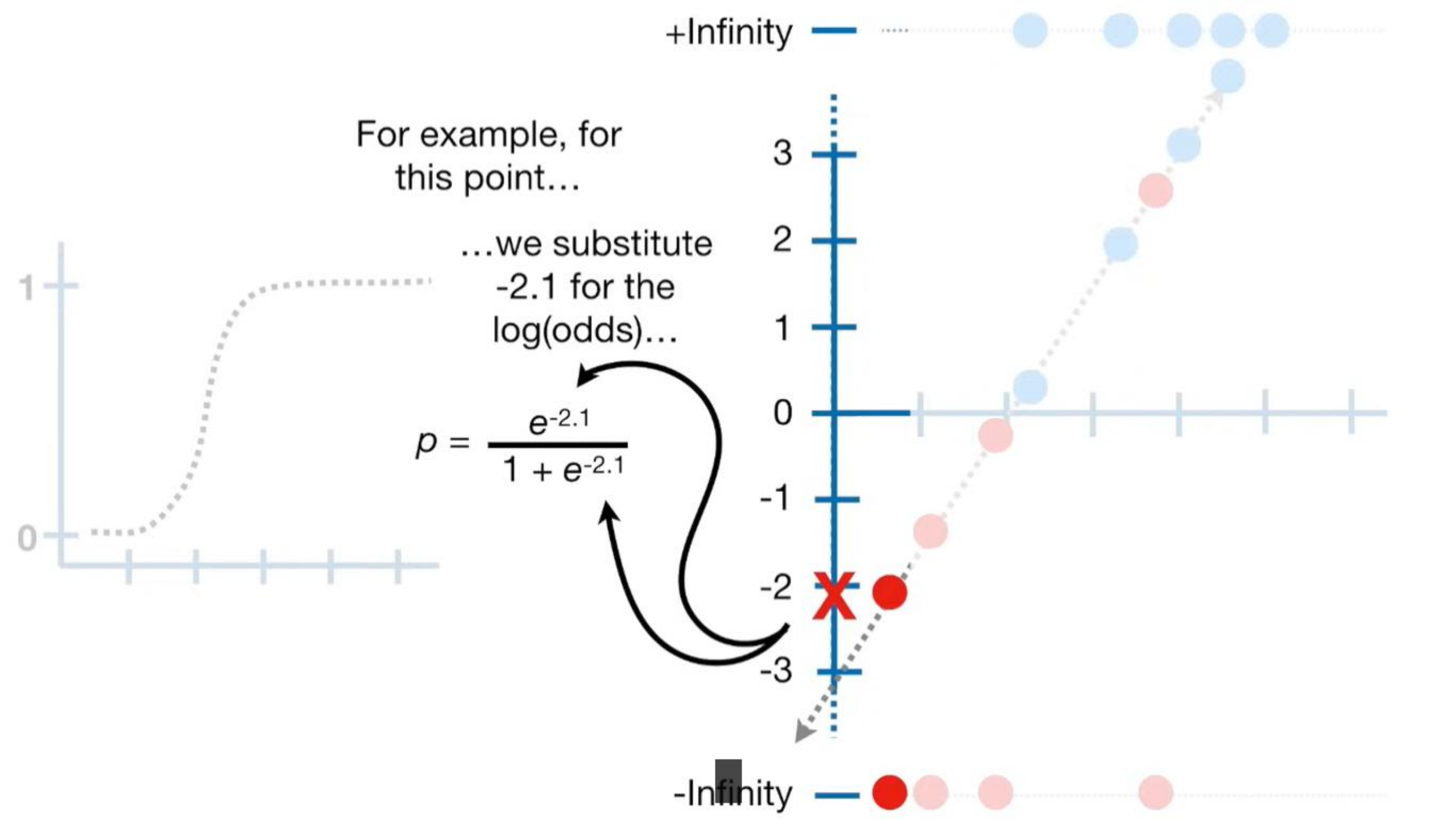


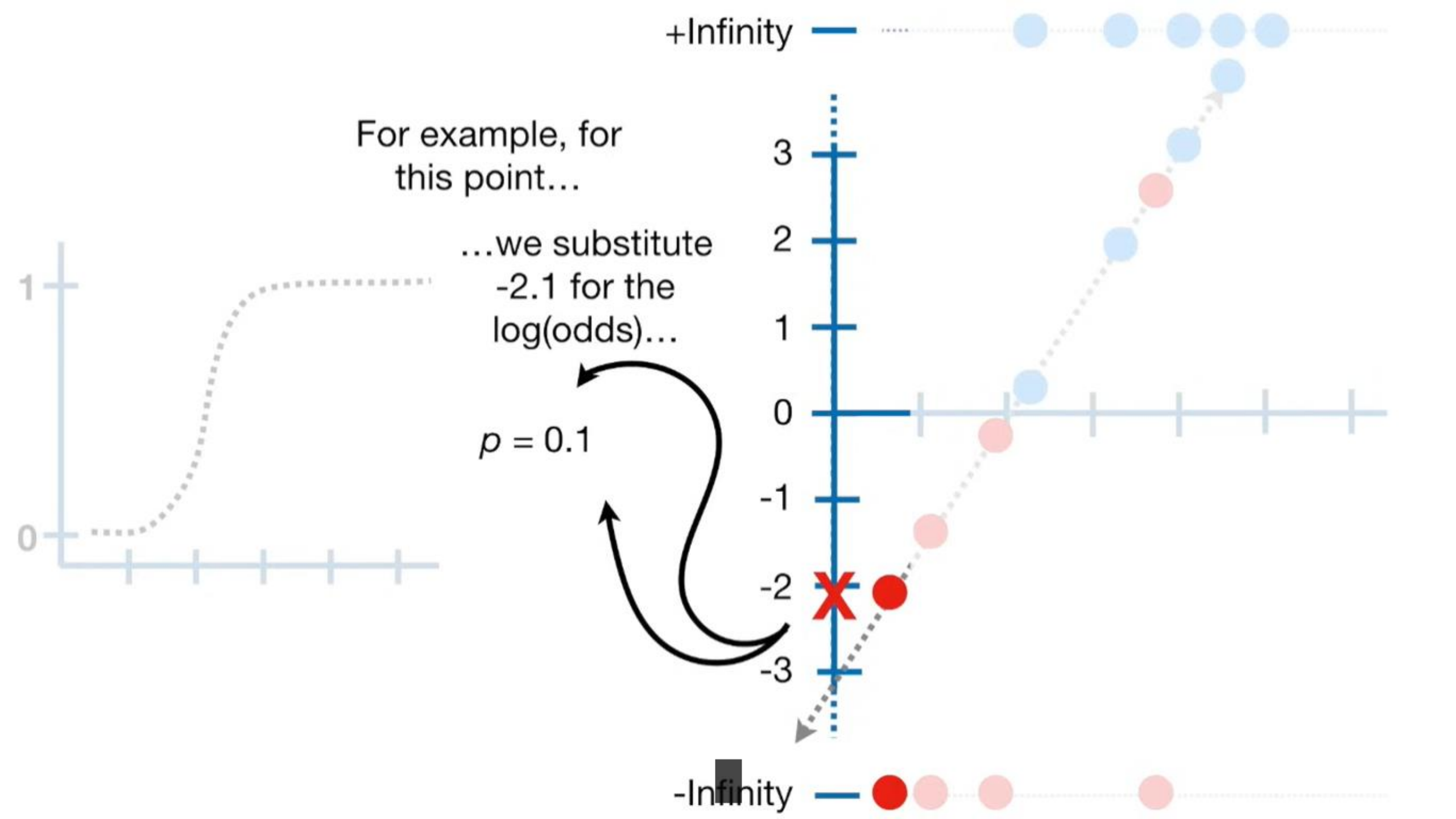
…which is just a reordering of the transformation from probability to log(odds).

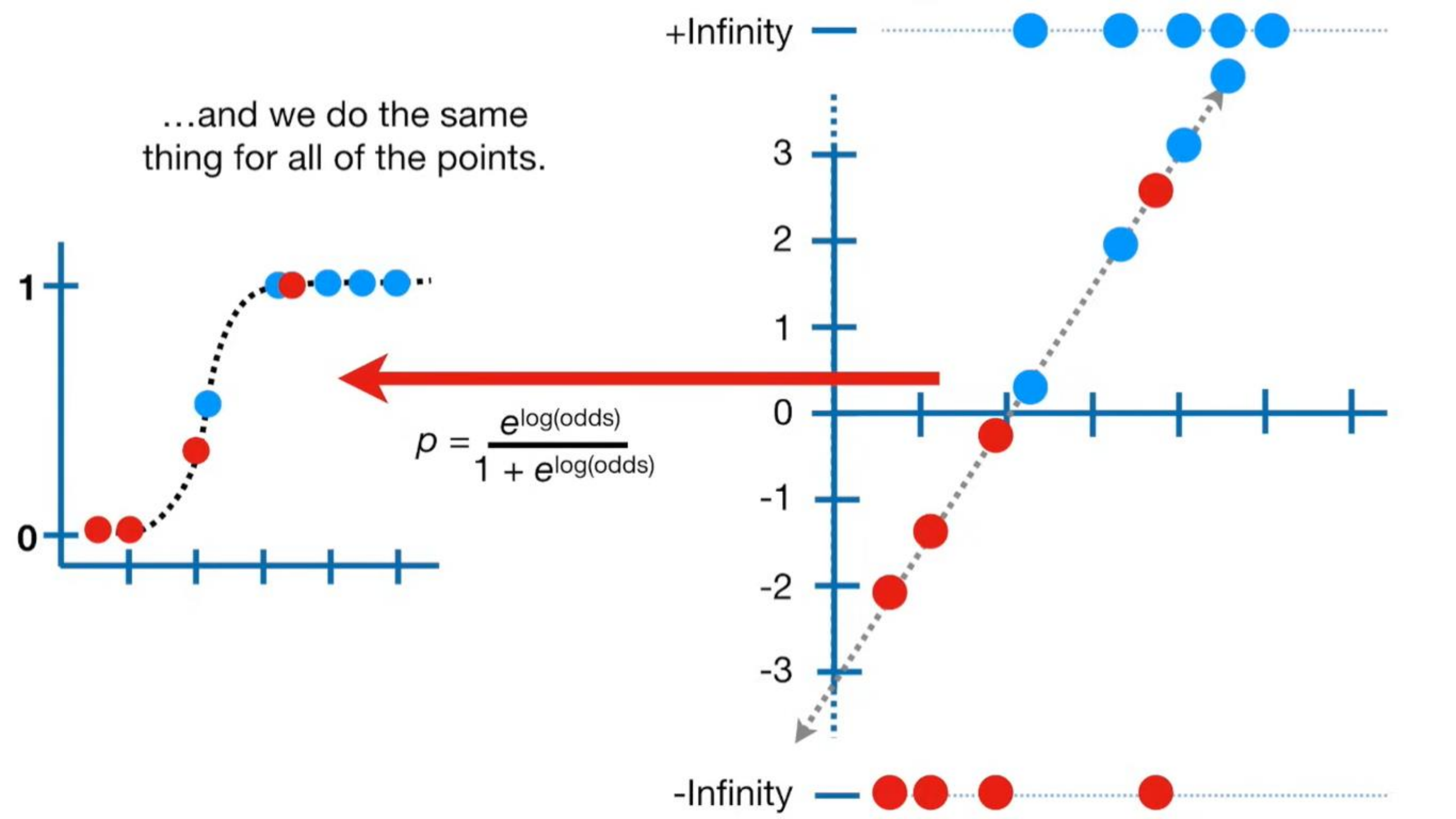


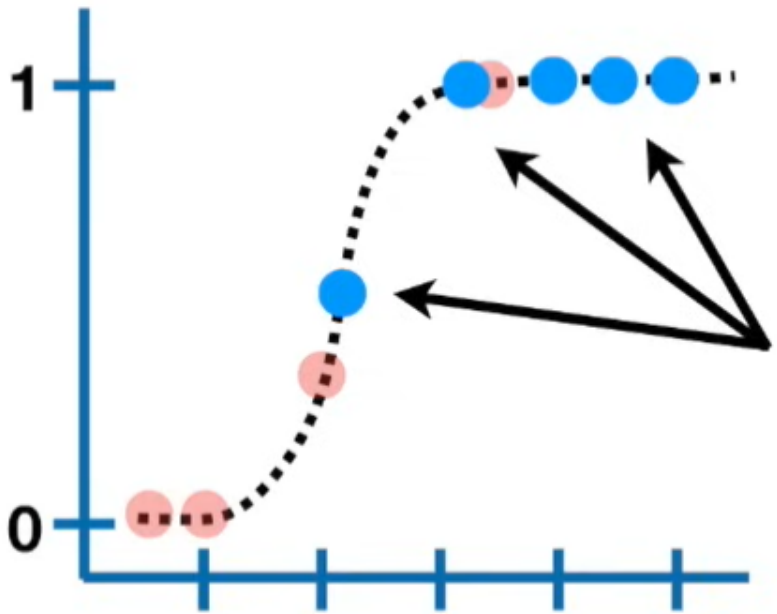
$$\log\left(\frac{p}{1-p}\right) = \log(\text{odds})$$



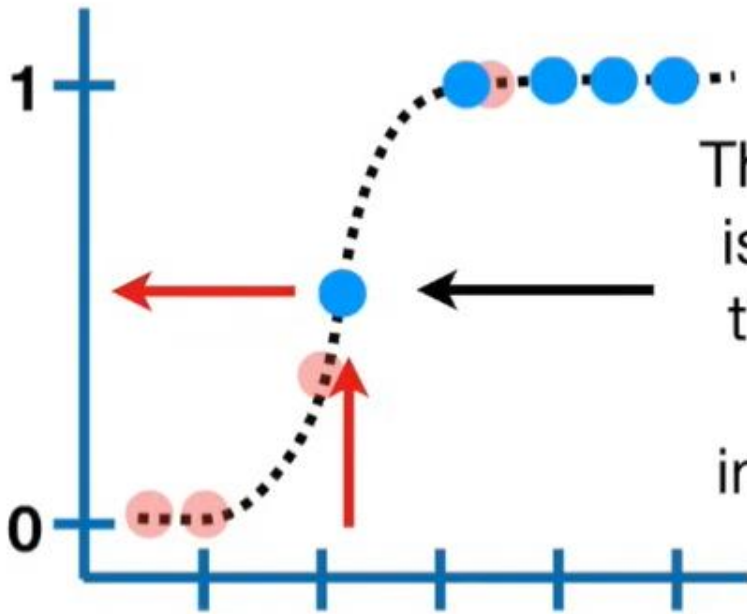

+Infinity
For example, for this point…
…we substitute -2.1 for the log(odds)…
$p = \frac{e^{-2.1}}{1 + e^{-2.1}}$
3
2
1
0
-1
-2
-3
1
0
-Infinity


+Infinity
For example, for this point…
…we substitute -2.1 for the log(odds)…
$p = 0.1$
3
2
1
0
-1
-2
-3
-Infinity
1
0

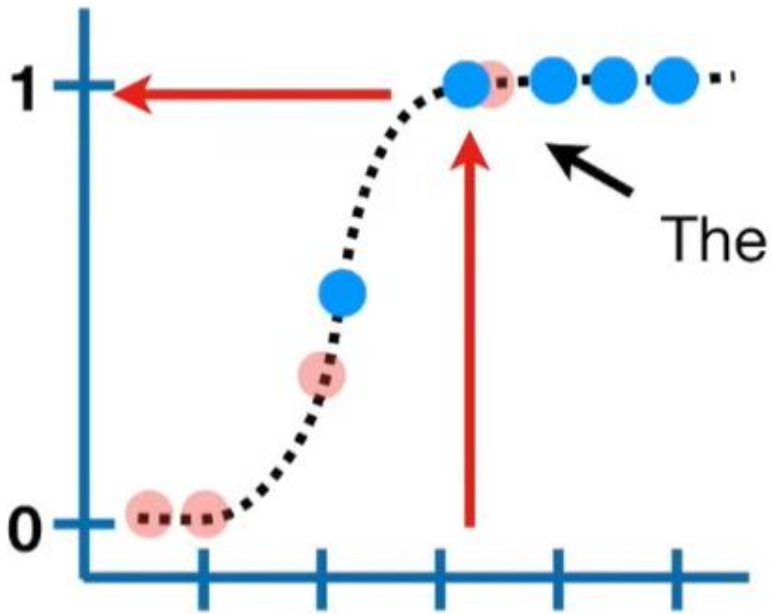

…and we do the same thing for all of the points.
1
0
$p = \frac{e^{\log(odds)}}{1 + e^{\log(odds)}}$
+Infinity
3
2
1
0
-1
-2
-3
-Infinity



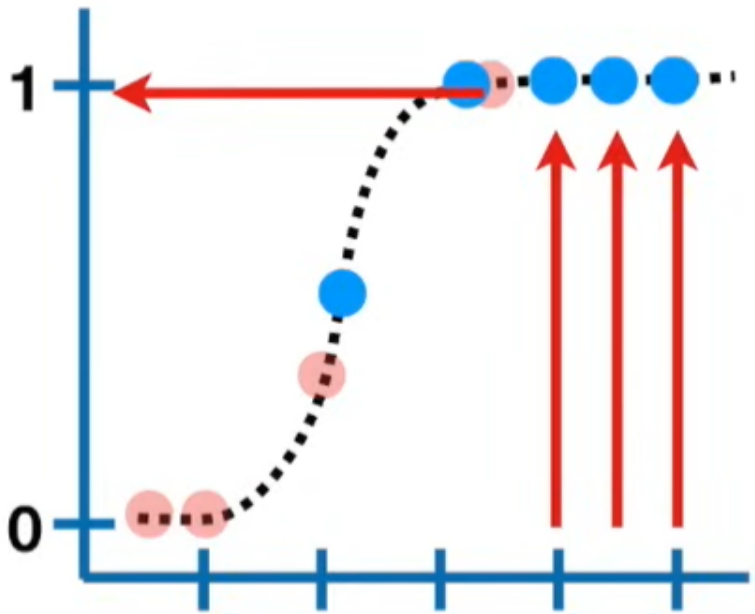
We'll start by calculating the likelihood of the **obese** mice, given the shape of the squiggle.



The likelihood that this mouse is **obese**, given the shape of the squiggle, is the value on the y-axis where point intersects the squiggle, **0.49**.

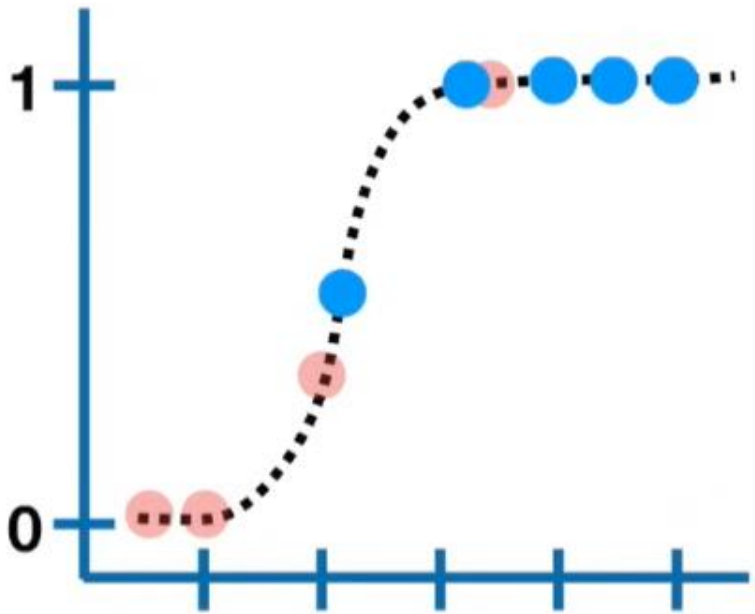


The likelihood that this mouse is **obese** is **0.9**



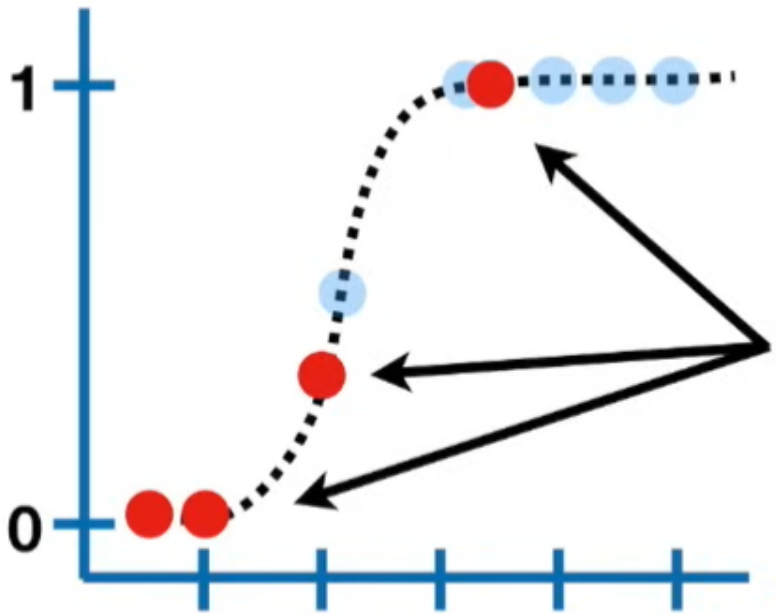
The likelihoods that these mice are **obese** are **0.91**, **0.91** and **0.92**

likelihood of data given the squiggle = **0.49** × **0.9** × **0.91** × **0.91** × **0.92** × …



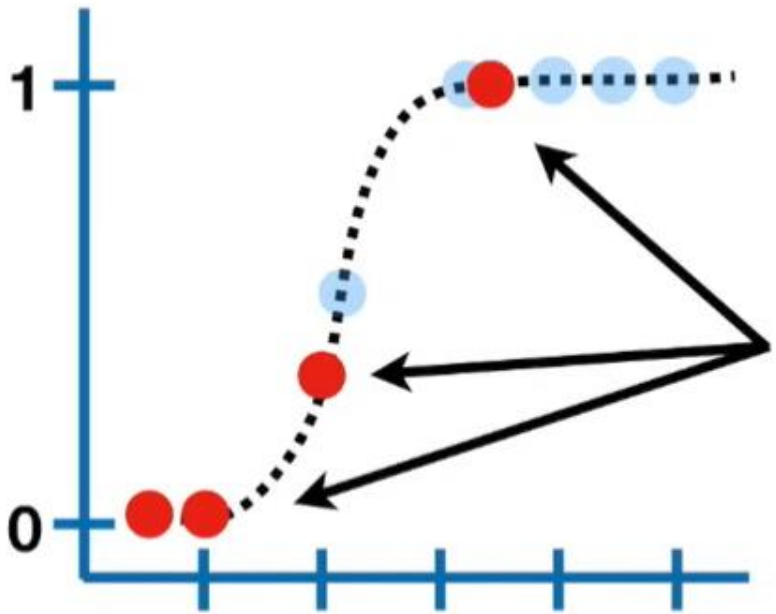
The likelihood for all of the **obese** mice is just the product of the individual likelihoods.

likelihood of data given the squiggle = 0.49 × 0.9 × 0.91 × 0.91 × 0.92 × …



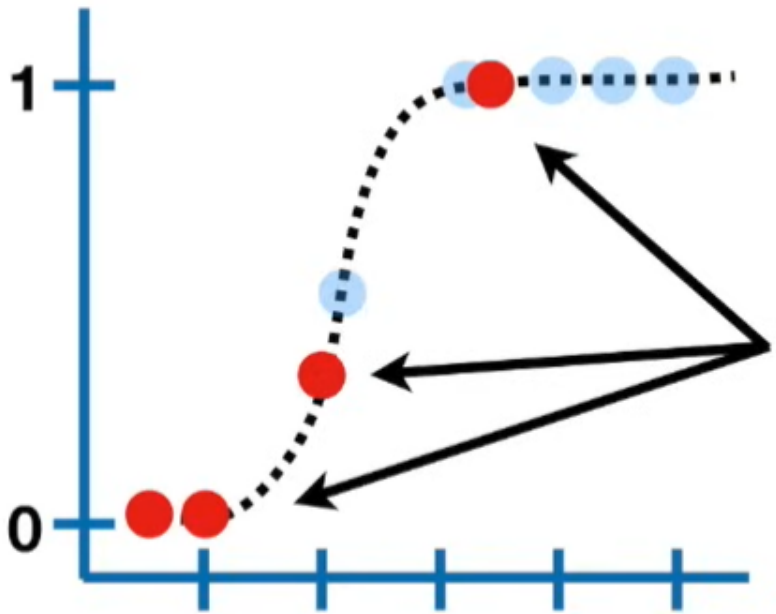
Now we'll figure out the likelihoods for the mice that are **not obese**.

likelihood of data given the squiggle = 0.49 × 0.9 × 0.91 × 0.91 × 0.92 × …



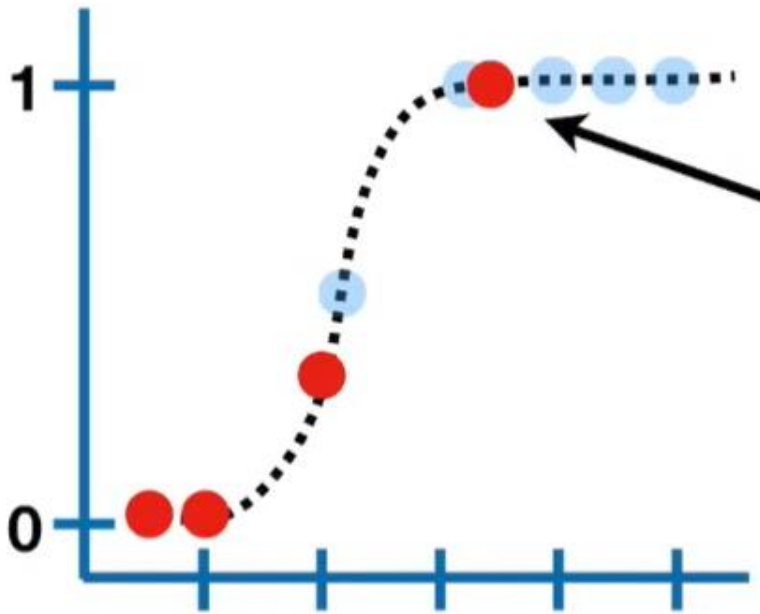
NOTE: The lower the probability of being obese, the higher the probability of not being obese.

likelihood of data given the squiggle = **0.49** x **0.9** x **0.91** x **0.91** x **0.92** x ...



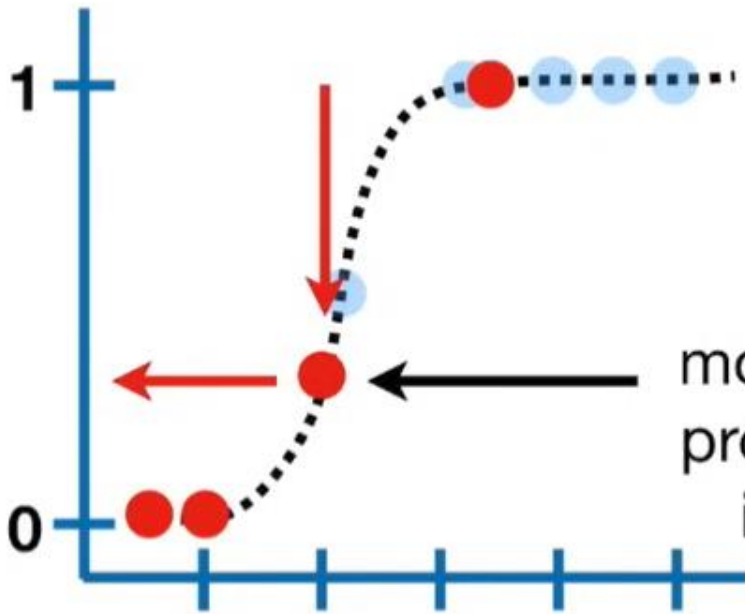
Thus, for these mice, the
likelihood = (1 - probability the mouse is **obese**)

likelihood of data given the squiggle = **0.49** × **0.9** × **0.91** × **0.91** × **0.92** × …



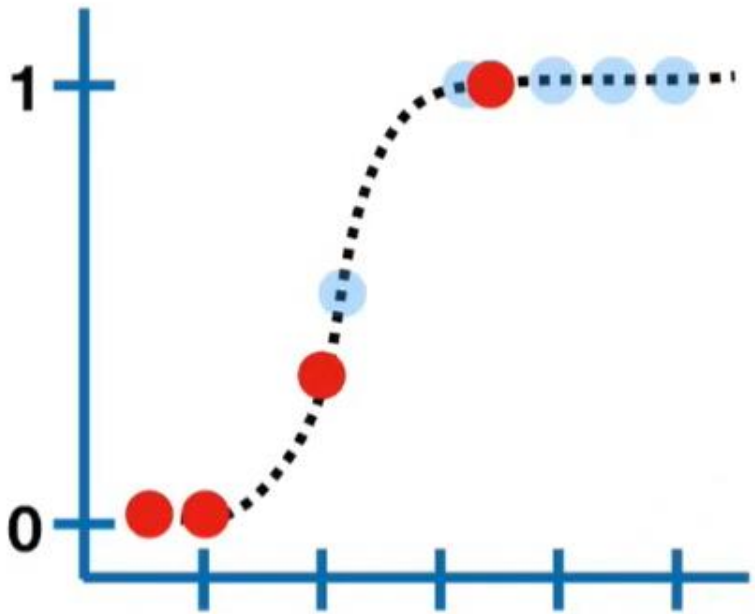
The probability that this mouse is **obese** is 0.9, so the probability and likelihood that it is **not obese** is (**1 - 0.9**)

likelihood of data given the squiggle = **0.49** × **0.9** × **0.91** × **0.91** × **0.92** × …



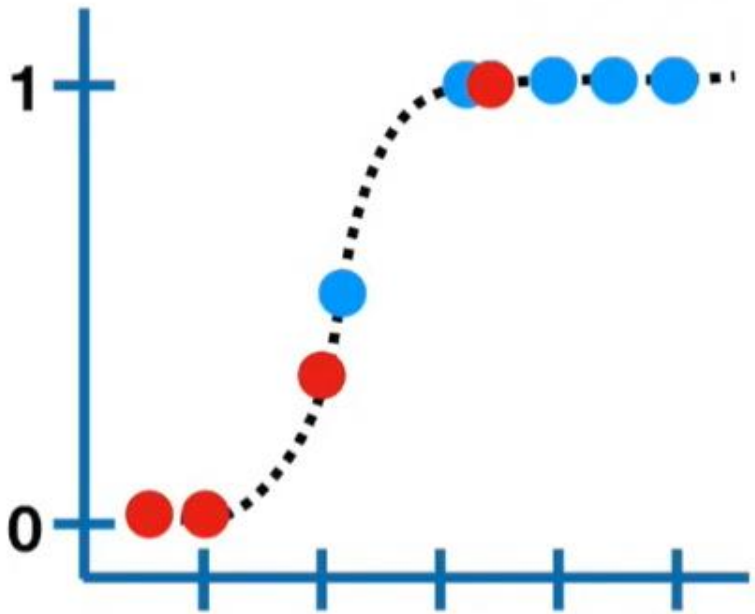
The probability that this mouse is **obese** is 0.3, so the probability and likelihood that it is **not obese** is (**1 - 0.3**)

likelihood of data given the squiggle = 0.49 × 0.9 × 0.91 × 0.91 × 0.92 ×
(1 - 0.9) × (1 - 0.3) × (1 - 0.01) × (1 - 0.01)



Now we can include the individual likelihoods for the mice that are **not obese** to the equation for the overall likelihood.

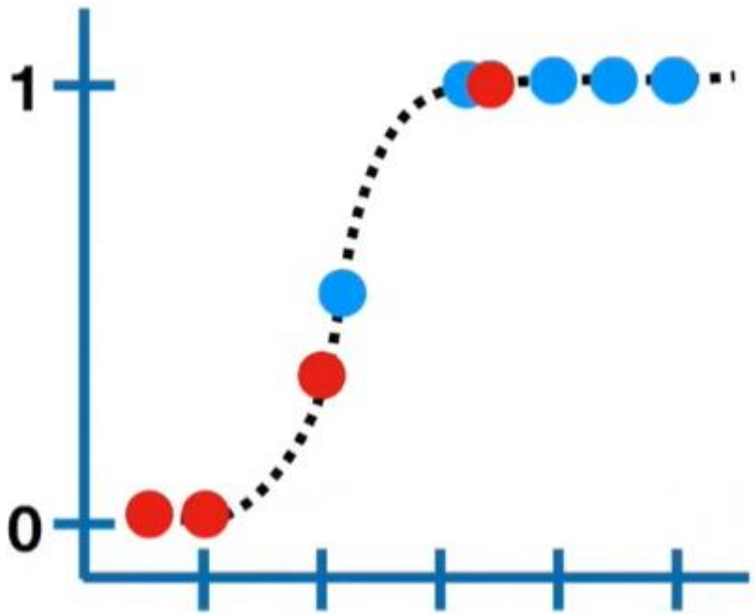
likelihood of data given the squiggle = $0.49 \times 0.9 \times 0.91 \times 0.91 \times 0.92 \times (1 - 0.9) \times (1 - 0.3) \times (1 - 0.01) \times (1 - 0.01)$



NOTE: Although it is possible to calculate the likelihood as the product of the individual likelihoods, statisticians prefer to calculate the **log of the likelihood** instead.

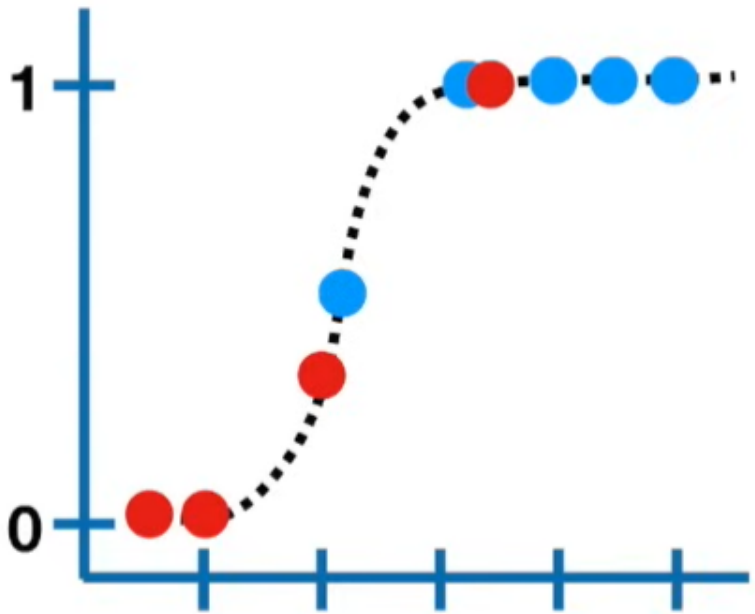
Either way works because the squiggle that maximizes the likelihood is the same one that maximizes the log of the likelihood.

log(likelihood of data given the squiggle) = log(0.49) + log(0.9) + log(0.91) + log(0.91) +
log(0.92) + log(1 - 0.9) + log(1 - 0.3) +
log(1 - 0.01) + log(1 - 0.01)



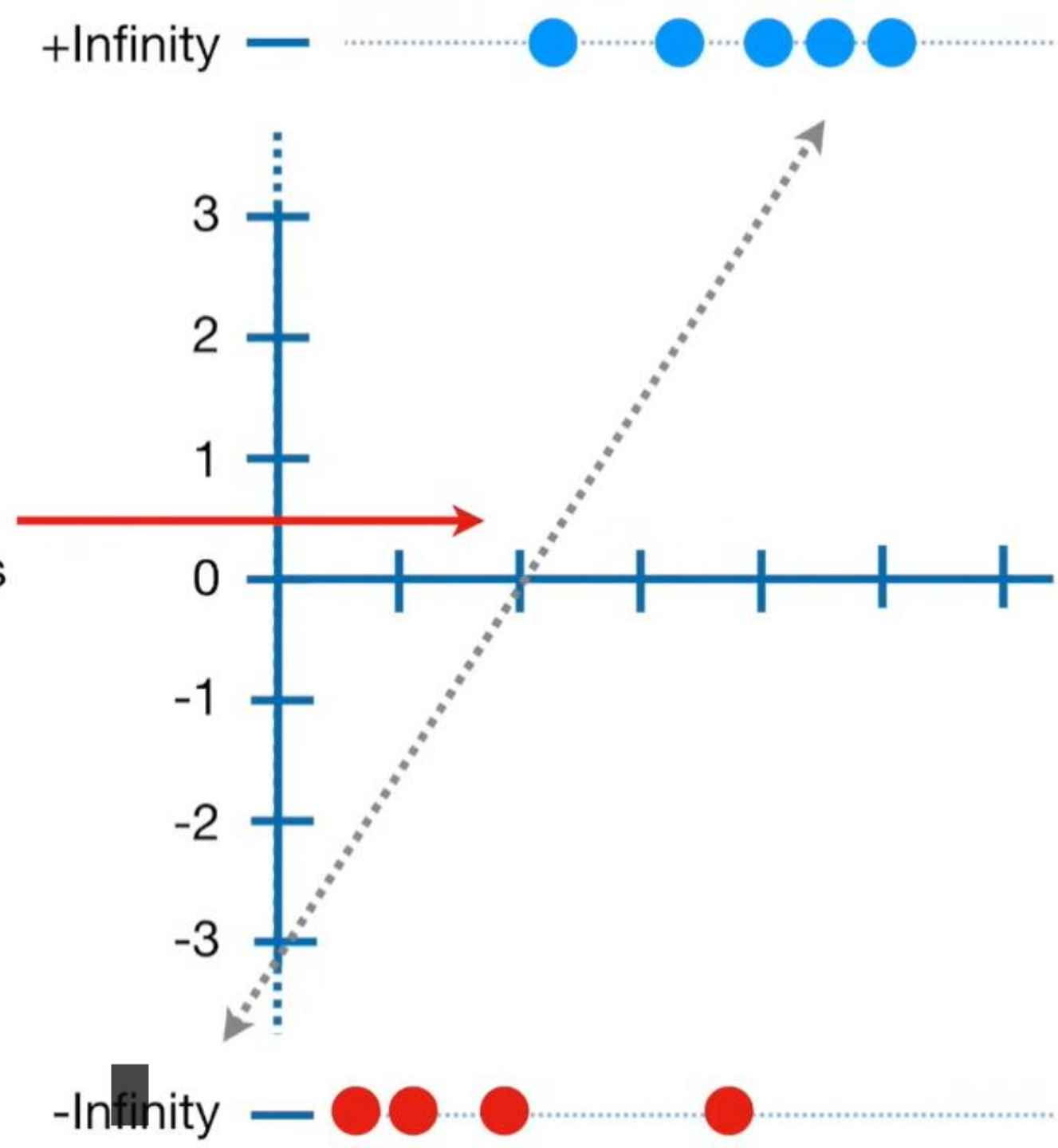
With the log of the likelihood, or
"log-likelihood" to those in the know, we
add the logs of the individual likelihoods
instead of multiplying the individual
likelihoods…

log(likelihood of data given the squiggle) = -3.77

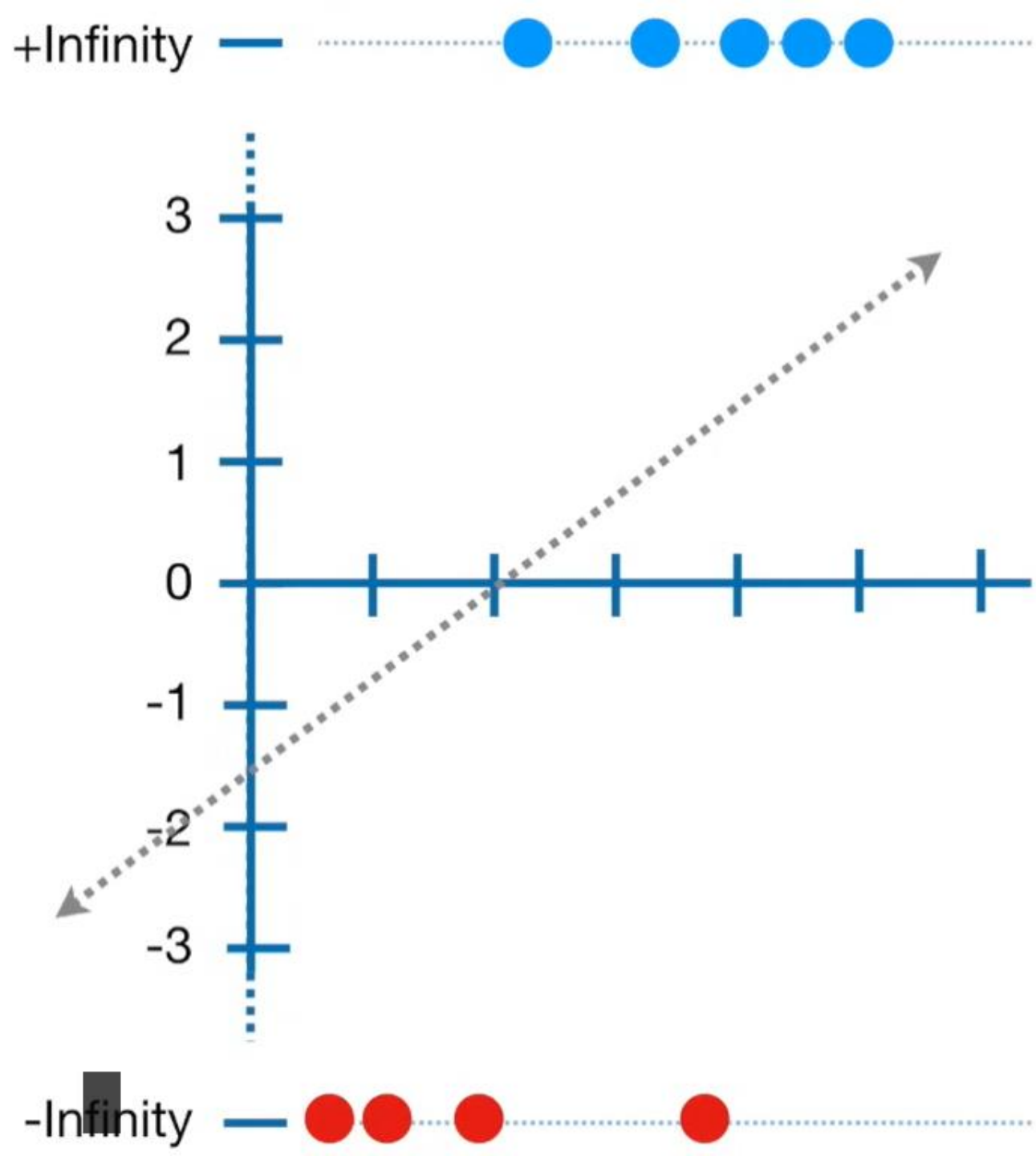


Thus, the log-likelihood of the data given the squiggle is -3.77…

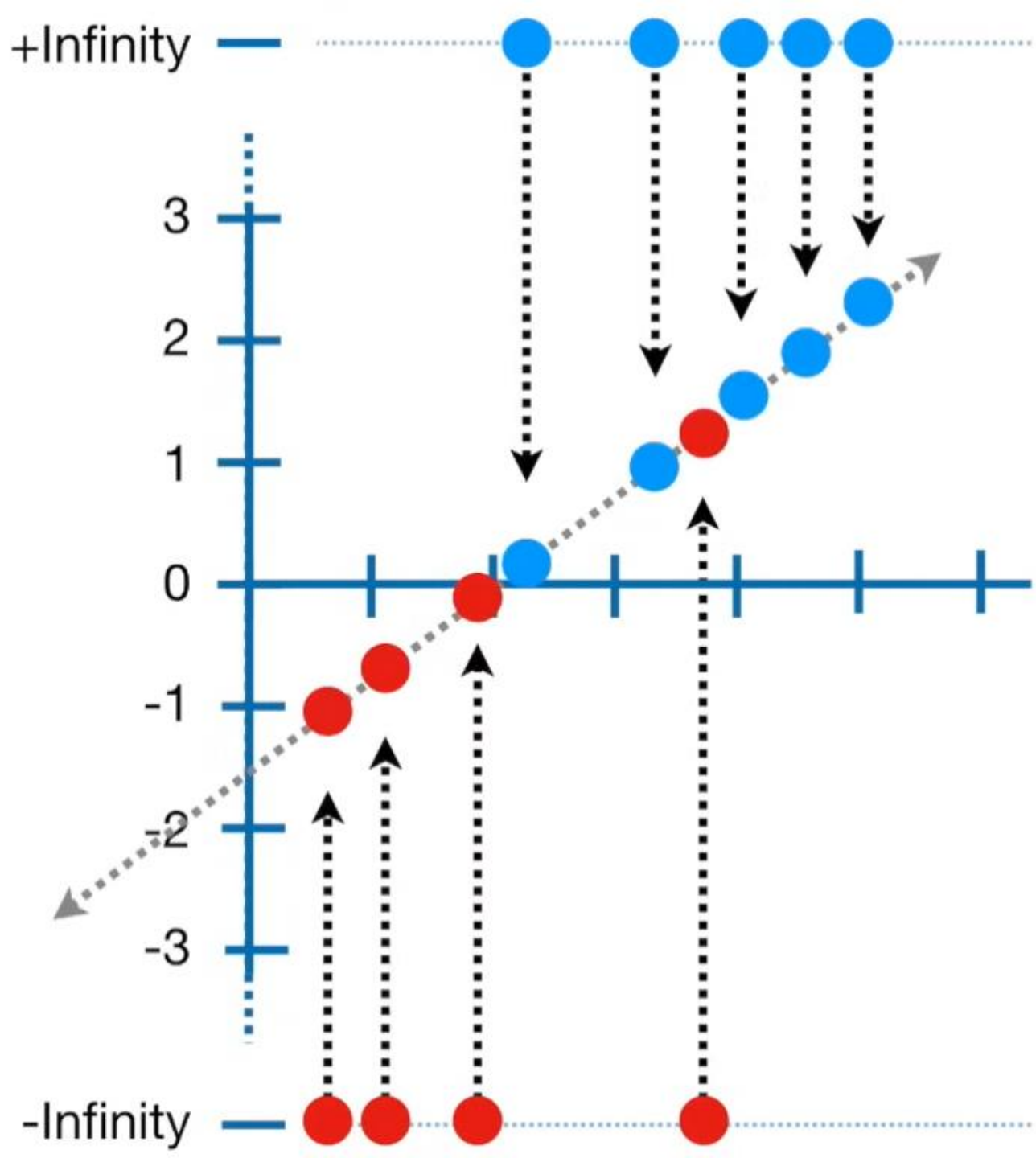
…and this means that the log-likelihood of the original line is -3.77.

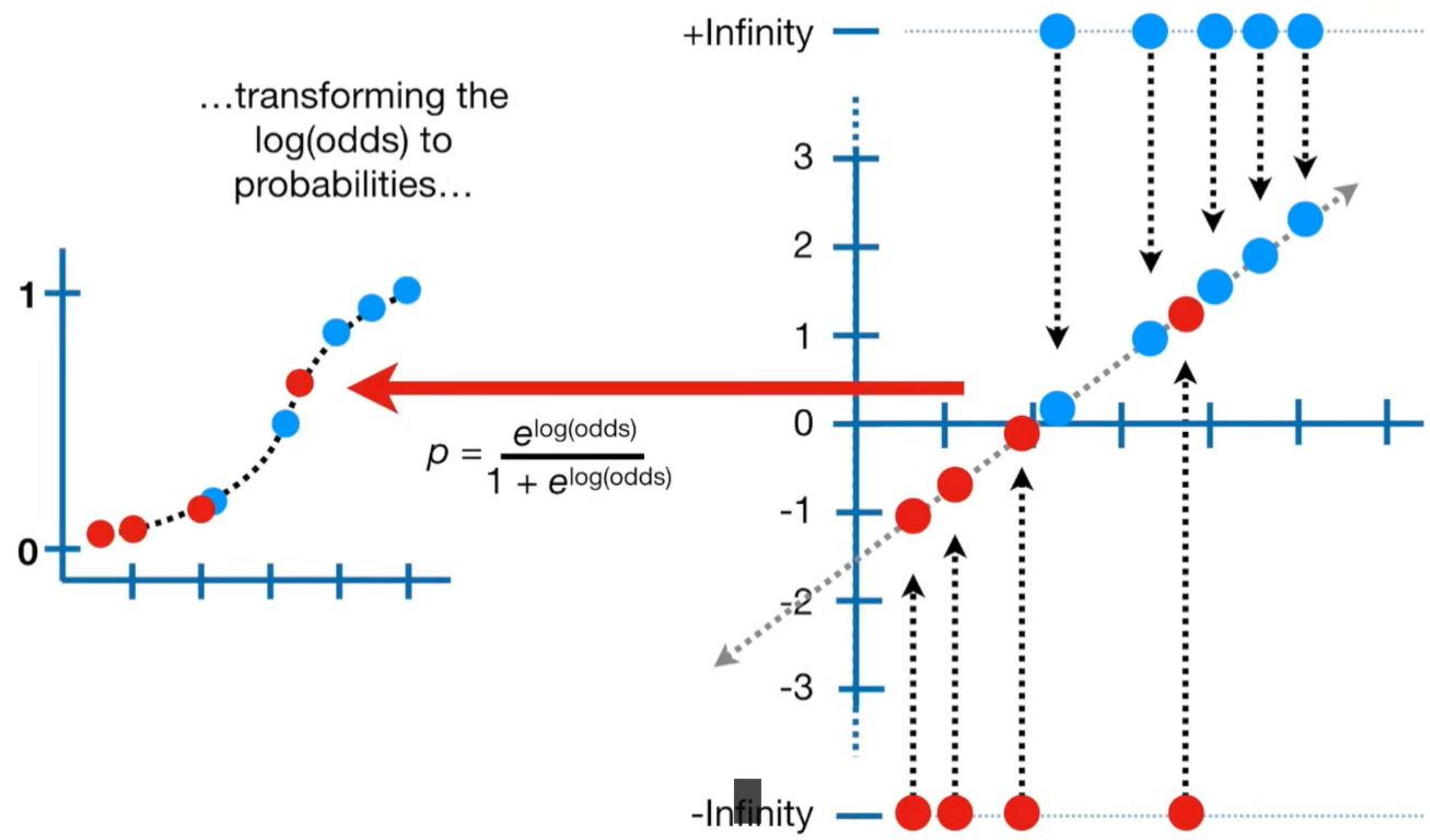


Now we rotate the line…

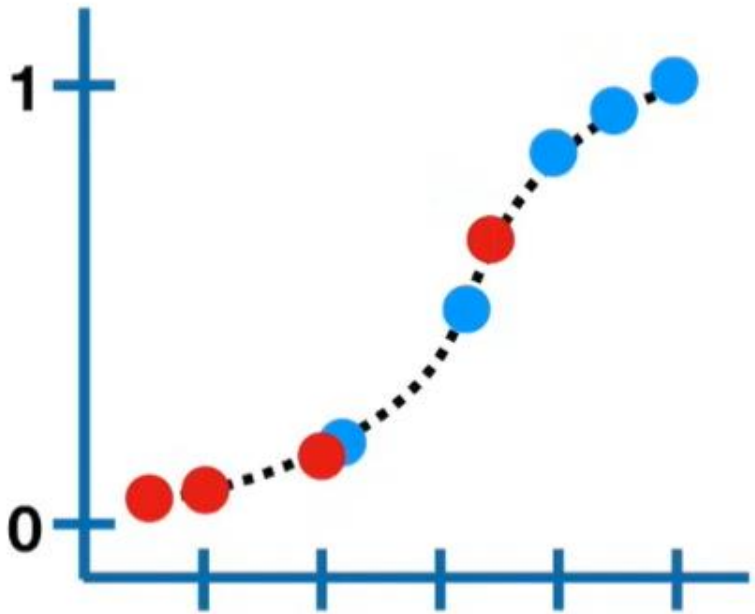


…and calculate its log-likelihood
by projecting the data onto it…




…transforming the log(odds) to probabilities…
$p = \frac{e^{\log(odds)}}{1 + e^{\log(odds)}}$
+Infinity
-Infinity
3
2
1
0
-1
-2
-3
1
0

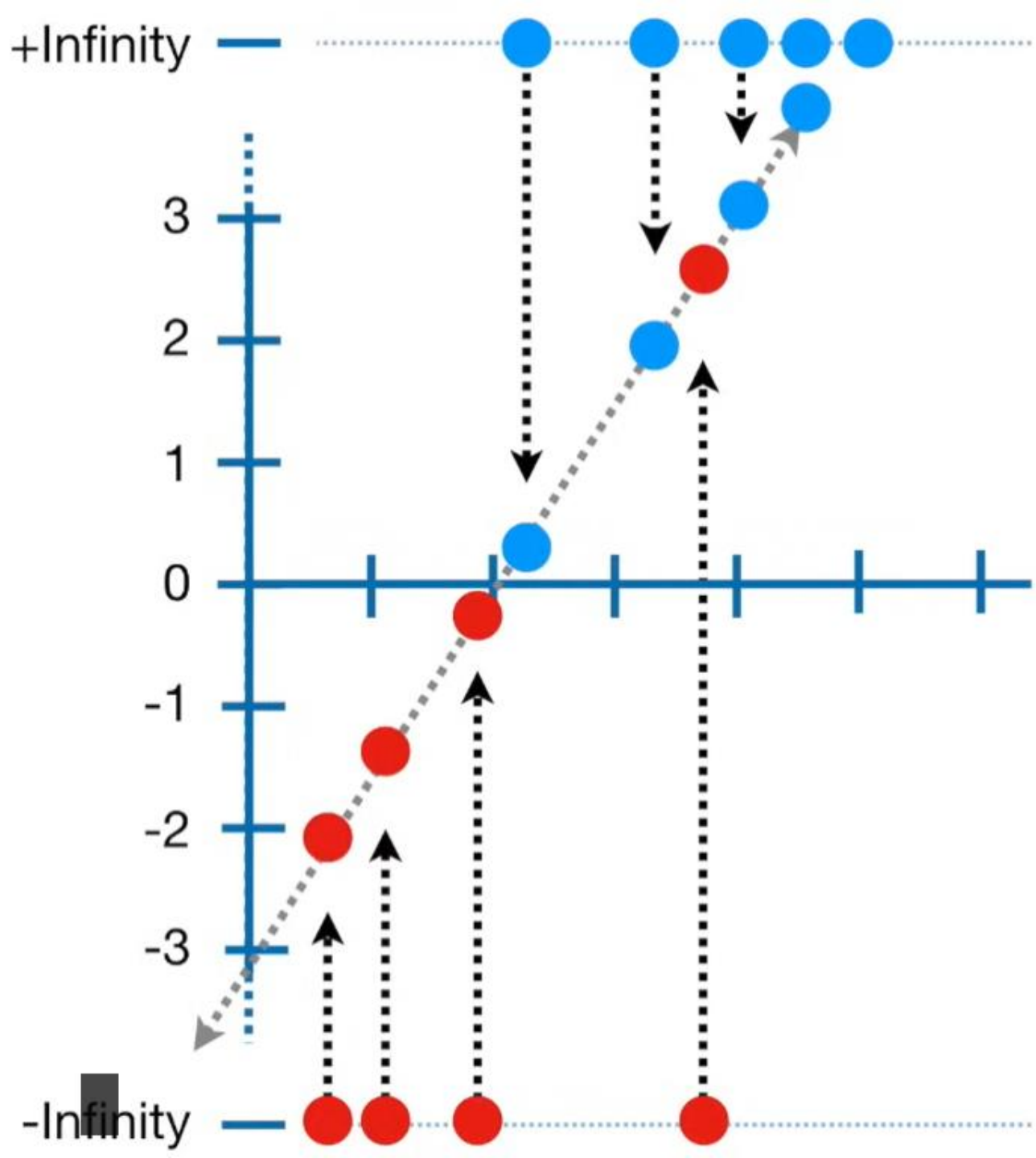
log(likelihood of data given the squiggle) = -4.15



…and the final value for the log-likelihood is -4.15.

(So this one is not as good as the first line.)

NOTE: The algorithm that finds the line with the maximum likelihood is pretty smart - each time it rotates the line, it does so in a way that increases the log-likelihood. Thus, the algorithm can find the optimal fit after a few rotations.



Maximum Likelihood

We use maximum likelihood to estimate the parameters.

$$\ell(\beta_0, \beta) = \prod_{i:y_i=1} p(x_i) \prod_{i:y_i=0} (1 - p(x_i)).$$

This *likelihood* gives the probability of the observed zeros and ones in the data. We pick β_0 and β_1 to maximize the likelihood of the observed data.

Most statistical packages can fit linear logistic regression models by maximum likelihood.

	Coefficient	Std. Error	Z-statistic	P-value
Intercept	-10.6513	0.3612	-29.5	< 0.0001
balance	0.0055	0.0002	24.9	< 0.0001

Making Predictions

What is our estimated probability of `default` for someone with a balance of \$1000?

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 X}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 X}} = \frac{e^{-10.6513+0.0055\times 1000}}{1+e^{-10.6513+0.0055\times 1000}} = 0.006$$

With a balance of \$2000?

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 X}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 X}} = \frac{e^{-10.6513+0.0055\times 2000}}{1+e^{-10.6513+0.0055\times 2000}} = 0.586$$

Lets do it again, using `student` as the predictor.

	Coefficient	Std. Error	Z-statistic	P-value
`Intercept`	-3.5041	0.0707	-49.55	< 0.0001
`student[Yes]`	0.4049	0.1150	3.52	0.0004

$$\widehat{\Pr}(\texttt{default=Yes}|\texttt{student=Yes}) = \frac{e^{-3.5041+0.4049\times 1}}{1+e^{-3.5041+0.4049\times 1}} = 0.0431,$$

$$\widehat{\Pr}(\texttt{default=Yes}|\texttt{student=No}) = \frac{e^{-3.5041+0.4049\times 0}}{1+e^{-3.5041+0.4049\times 0}} = 0.0292.$$

Logistic Regression with multiple predictors

- We now consider the problem of predicting a binary response using multiple predictors.

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}.$$

we use the maximum likelihood method to estimate $\beta_0, \beta_1, \ldots, \beta_p$.

- $\log(\frac{p}{1-p})=X^T\beta= \beta_0+ \beta_1x_1+\ldots+ \beta_px_p$
- $Pr(y_i=1|x_i)=\frac{e^{x_i^T\beta}}{1+e^{x_i^T\beta}}=p_i$
- $Pr(y_i=0|x_i)=1- \frac{e^{x_i^T\beta}}{1+e^{x_i^T\beta}}=\frac{1}{1+e^{x_i^T\beta}}=1- p_i$
- Likelihood of a single observation (y_i,x_i) is $f(y_i,x_i|\ \beta)$

$$=(p_i)^{y_i}(1-p_i)^{(1-y_i)}$$

$$=(\frac{e^{x_i^T\beta}}{1+e^{x_i^T\beta}})^{y_i}(\frac{1}{1+e^{x_i^T\beta}})^{(1-y_i)}$$

- Joint Likelihood of all observation assuming observations are independent is ι(β: y,x)= $\prod_{i=1}^{n}(p_i{}^{y_i}(1-p_i)^{(1-y_i)})$
- Maximum likelihood estimate of β is

$$\hat{\beta} = \underset{\beta}{argmax}\ \iota(\beta: y,x)$$

$$\hat{\beta} = \underset{\beta}{argmax}\ \log(\iota(\beta: y,x))$$

$$\hat{\beta} = \underset{\beta}{argmin}(-\log(\iota(\beta: y,x)))$$

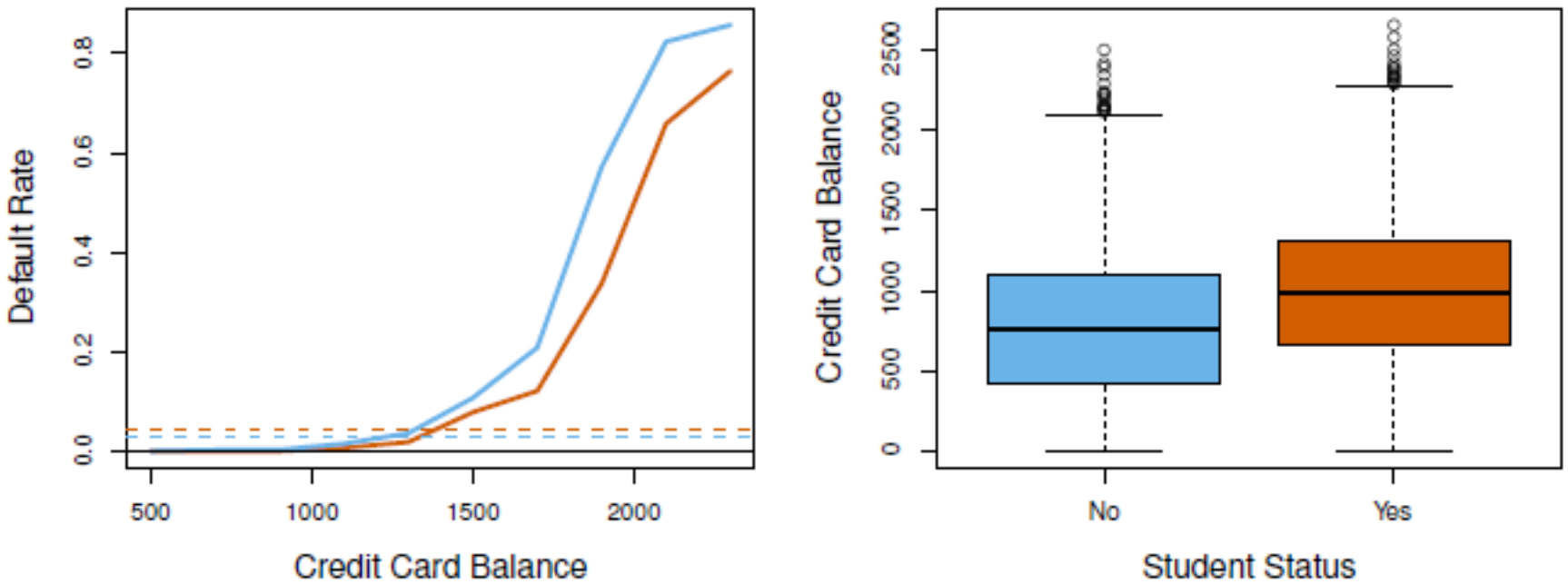
Gradient descent function can be used to minimize negative log likelihood function.

Example with multiple predictors

	Coefficient	Std. Error	Z-statistic	P-value
`Intercept`	-10.8690	0.4923	-22.08	< 0.0001
`balance`	0.0057	0.0002	24.74	< 0.0001
`income`	0.0030	0.0082	0.37	0.7115
`student[Yes]`	-0.6468	0.2362	-2.74	0.0062

Why is coefficient for `student` negative, while it was positive before?

Confounding



- Students tend to have higher balances than non-students, so their marginal default rate is higher than for non-students.
- But for each level of balance, students default less than non-students.
- Multiple logistic regression can tease this out.

- For example, a student with a credit card balance of $1, 500 and an income of $40, 000 has an estimated probability of default of

$$\hat{p}(X) = \frac{e^{-10.869+0.00574\times 1{,}500+0.003\times 40-0.6468\times 1}}{1+e^{-10.869+0.00574\times 1{,}500+0.003\times 40-0.6468\times 1}} = 0.058.$$

- A non-student with the same balance and income has an estimated probability of default of

$$\hat{p}(X) = \frac{e^{-10.869+0.00574\times 1{,}500+0.003\times 40-0.6468\times 0}}{1+e^{-10.869+0.00574\times 1{,}500+0.003\times 40-0.6468\times 0}} = 0.105.$$

Multi-Class classification using Logistic Regression

- Multi-class classification using logistic regression can be achieved through several approaches, with one of the most common being the "one-vs-all" (also known as "one-vs-rest") method.
- **Model Training**: Train a separate binary logistic regression classifier for each class in your dataset. For each classifier, the goal is to distinguish that class from all other classes (hence the "one-vs-all" approach).
- **Prediction**:
- a. For a new instance, apply each binary classifier to obtain a probability that the instance belongs to each class.
- b. The class associated with the highest probability becomes the predicted class for the instance.

Why the name Logistic Regression?

- The name "logistic regression" may seem somewhat misleading at first glance because it contains the term "regression," which typically implies a model used for predicting continuous outcomes.
- The term "logistic" in logistic regression refers to the logistic function, also known as the sigmoid function, which is used to transform the output of the linear equation into a probability score between 0 and 1.
- The "regression" part of the name comes from the fact that logistic regression is based on a linear regression model. In logistic regression, we start with a linear combination of the predictor variables, similar to linear regression. However, instead of predicting a continuous outcome directly, logistic regression predicts the probability that an instance belongs to a particular class.

Logistic Regression might exhibit instability

- **Collinearity**: Logistic regression can be sensitive to multicollinearity, where predictor variables are highly correlated.
- **Outliers**: Logistic regression can also be affected by outliers, especially if they are influential points that exert disproportionate influence on the estimated coefficients.
- **Small Sample Sizes**: In situations with small sample sizes, logistic regression estimates may be less stable due to increased variability in the parameter estimates.
- **Imbalanced Data**: In the case of highly imbalanced datasets where one class is significantly more prevalent than the other, logistic regression may produce unstable estimates.

Linear Discriminant Analysis (LDA)

- Logistic regression involves directly modeling $Pr(Y = k|X = x)$ using the logistic function.
- LDA considers an alternative and less direct approach to estimating these probabilities.
- In this alternative approach, we model the distribution of the predictors *X* separately in each of the response classes (i.e. given *Y*).
- Then use Bayes' theorem to flip these around into estimates for $Pr(Y = k|X = x)$.

Bayes theorem for classification

Thomas Bayes was a famous mathematician whose name represents a big subfield of statistical and probabilistic modeling. Here we focus on a simple result, known as Bayes theorem:

$$\Pr(Y = k|X = x) = \frac{\Pr(X = x|Y = k) \cdot \Pr(Y = k)}{\Pr(X = x)}$$

One writes this slightly differently for discriminant analysis:

$$\Pr(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}, \quad \text{where}$$

- $f_k(x) = \Pr(X = x|Y = k)$ is the *density* for X in class k. Here we will use normal densities for these, separately in each class.
- $\pi_k = \Pr(Y = k)$ is the marginal or *prior* probability for class k.

- Let $f_k(X) \equiv \Pr(X = x | Y = k)$ denotes the *density function* of X for an observation that comes from the kth class. In other words, $f_k(X)$ is relatively large if there is a high probability that an observation in the kth class has $X \approx x$, and $f_k(X)$ is small if it is very unlikely that an observation in the kth class has $X \approx x$.
- we will use the abbreviation $p_k(X) = \Pr(Y = k|X)$ and referred as posterior probability
- If we can simply plug in estimates of π_k and $f_k(X)$ into following equation then we can estimate $p_k(X)$

$$\Pr(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)} \quad \text{-----Eq (1)}$$

- In general, estimating π_k is easy if we have a random sample of Y s from the population: we simply compute the fraction of the training observations that belong to the kth class.
- However, estimating $f_k(X)$ tends to be more challenging, unless we assume some simple forms for these densities.

Linear Discriminant Analysis for p = 1

- For now, assume that $p = 1$—that is, we have only one predictor.
- We would like to obtain an estimate for $f_k(x)$ that we can plug into Eq (1) in order to estimate $p_k(x)$.
- We will then classify an observation to the class k for which $p_k(x)$ is greatest.
- Suppose we assume that $f_k(x)$ is *normal* or *Gaussian*.

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left(-\frac{1}{2\sigma_k^2}(x-\mu_k)^2\right)$$

- where μ_k and σ_k^2 are the mean and variance parameters for the kth class.
- For now, let us further assume that $\sigma_1^2 = \ldots = \sigma_k^2$: that is, there is a shared variance term across all K classes, which for simplicity we can denote by σ^2.

$$p_k(x) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x-\mu_k)^2\right)}{\sum_{l=1}^{K} \pi_l \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x-\mu_l)^2\right)}$$

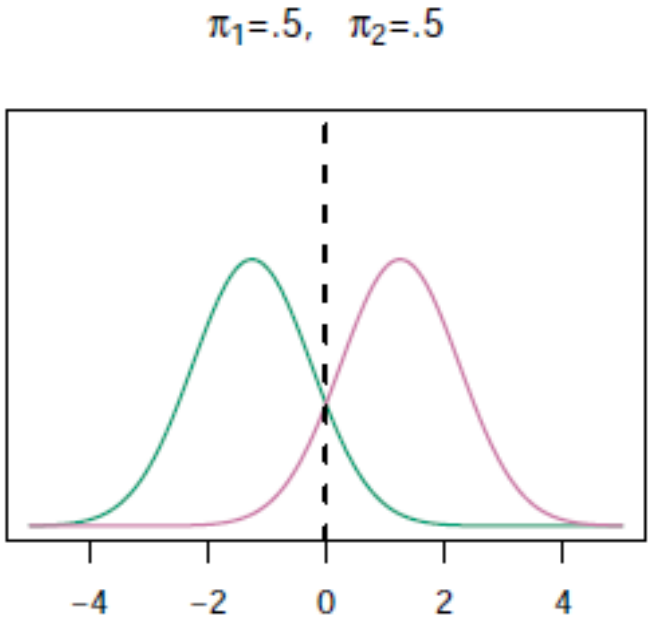
- Taking the log of last equation and ignoring those terms which does not involve k

$$\delta_k(x) = x \cdot \frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log(\pi_k) \qquad \text{--Eq (2)}$$

it is not hard to show that this is equivalent to assigning the observation to the class for which $\delta_k(x)$ is the largest.

- For instance, if $K = 2$ and $\pi_1 = \pi_2$ then
- $\delta_1(x) - \delta_2(x) = x.\frac{\mu_1}{\sigma^2} - \frac{{\mu_1}^2}{2\sigma^2} - x.\frac{\mu_2}{\sigma^2} + \frac{{\mu_2}^2}{2\sigma^2} = \frac{1}{2\sigma^2}(2x(\mu_1 - \mu_2) - ({\mu_1}^2 - {\mu_2}^2))$

  $\Rightarrow \delta_1(x) > \delta_2(x)$ when $2x(\mu_1 - \mu_2) > ({\mu_1}^2 - {\mu_2}^2)$
- In this case, the Bayes decision boundary corresponds to the point where
- $X = \frac{{\mu_1}^2 - {\mu_2}^2}{2(\mu_1 - \mu_2)} = \frac{\mu_1 + \mu_2}{2}$ ---(Eq 3)



- The two normal density functions that are displayed, $f_1(x)$ and $f_2(x)$, represent two distinct classes.
- The mean and variance parameters for the two density functions are $\mu_1 = -1.25$, $\mu_2 = 1.25$, and $\sigma_1^2 = \sigma_2^2 = 1$.
- We also assumes $\pi_1 = \pi_2 = 0.5$
- by inspection of Eq (3), we see that the Bayes classifier assigns the observation to class 1 if $x < 0$ and class 2 otherwise.

- Note that in this case, we can compute the Bayes classifier because we know that X is drawn from a Gaussian distribution within each class, and we know all of the parameters involved.
- In practice, even if we are quite certain of our assumption that X is drawn from a Gaussian distribution within each class, we still have to estimate the parameters $\mu_1, \ldots, \mu_K, \pi_1, \ldots, \pi_K$, and σ^2.
- The *linear discriminant analysis* (LDA) method approximates the Bayes classifier by plugging estimates for π_k, μ_K, and σ^2 into Eq (2).

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i:y_i=k} x_i$$

$$\hat{\sigma}^2 = \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i:y_i=k} (x_i - \hat{\mu}_k)^2$$

$$\hat{\pi}_k = n_k / n.$$

where n is the total number of training observations, and n_k is the number of training observations in the kth class.

Why discriminant analysis?

- When the classes are well-separated, the parameter estimates for the logistic regression model are surprisingly unstable. Linear discriminant analysis does not suffer from this problem.
- If n is small and the distribution of the predictors X is approximately normal in each of the classes, the linear discriminant model is again more stable than the logistic regression model.
- Linear discriminant analysis is popular when we have more than two response classes, because it also provides low-dimensional views of the data.