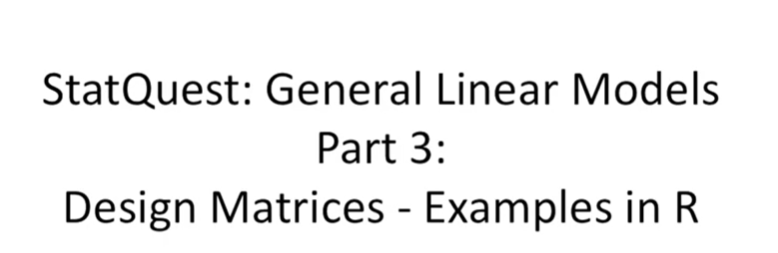
<https://www.youtube.com/watch?v=Hrr2anyK_5s&list=PLblh5JKOoLUIzaEkCLIUxQFjPIlapw8nU&index=8>



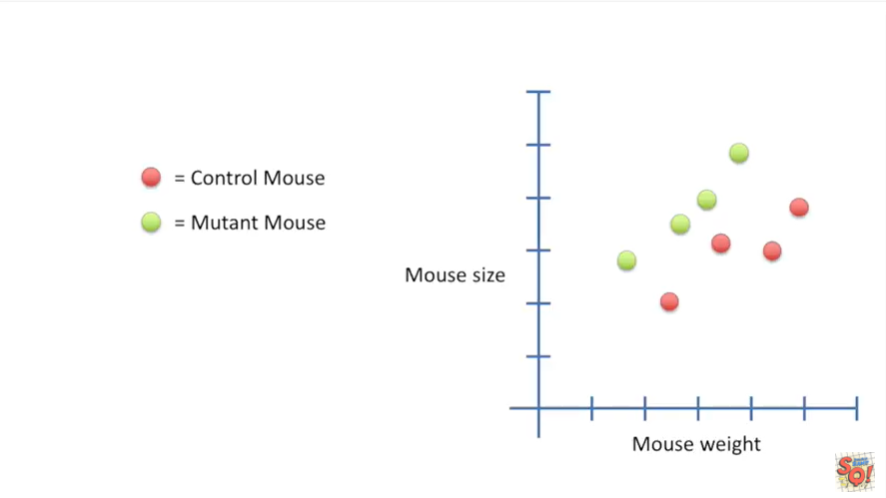
Hello and welcome to stat quest.

Stat quest is brought to you by the friendly folks in the genetics department at the University of North Carolina at Chapel Hill.

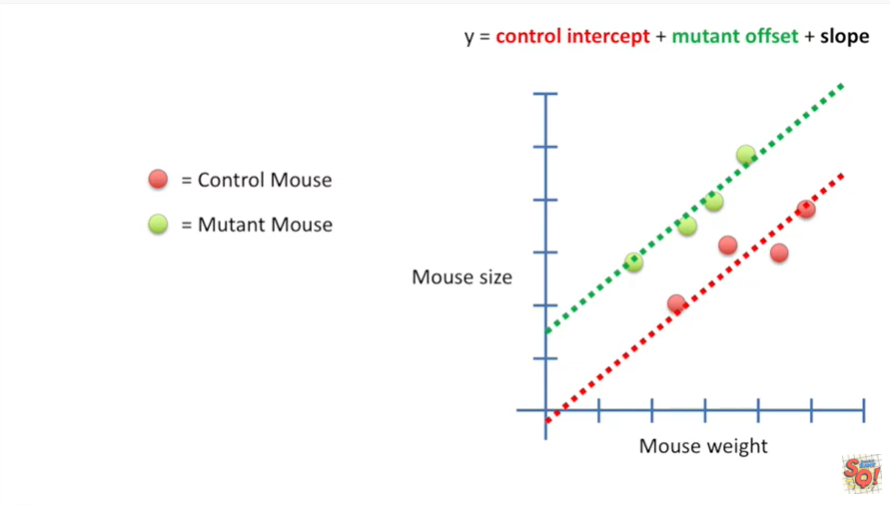
This stat quest complements the stat quest on general linear models part three

The one that focused on design matrices.

In this video I show you how to do the examples from the original stat quest in R.



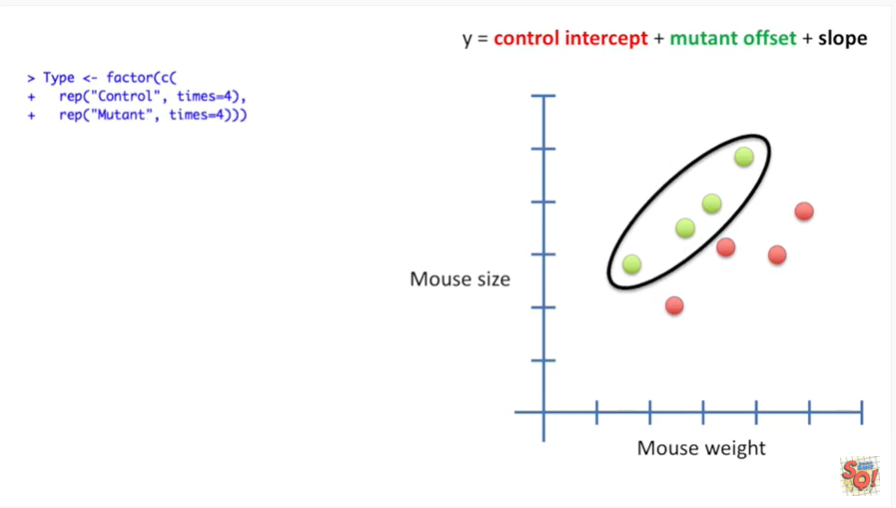
In the original stat quest we were interested in determining if there was a statistically significant difference between the size of control mice versus the size of mutant mice given that we have measured their weights.



And we ended up with this equation y equals the control intercept plus a mutant offset plus the slope.

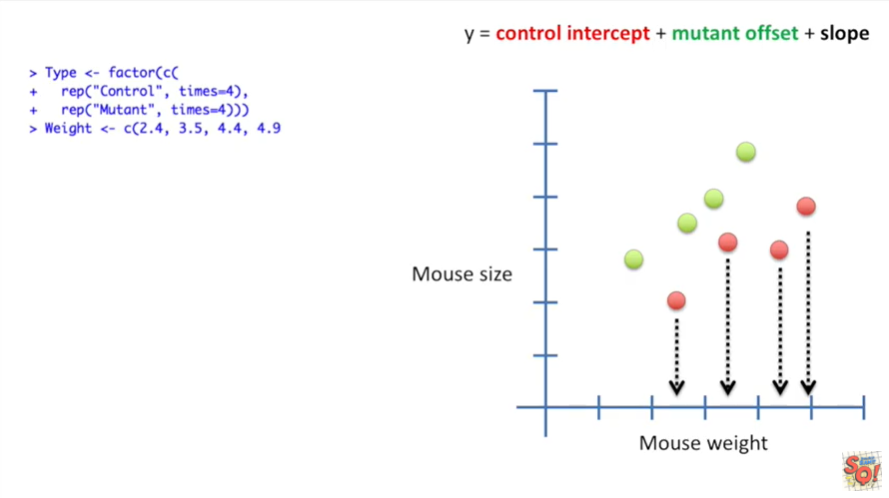
This equation fits two lines to the data one for the control mice and one for the mutant mice.

Now let's see how we can test the hypothesis that there's no difference between control mice and mutant mice using R.

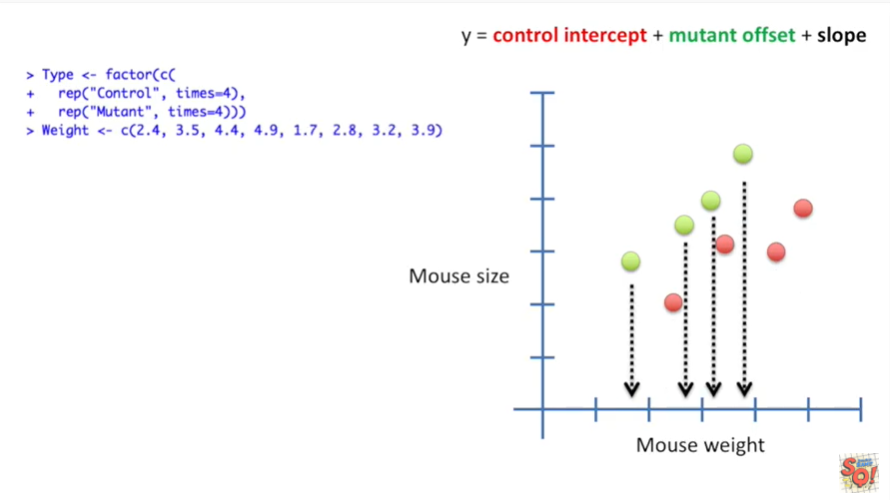


The first thing we do is we create labels for the control mice.

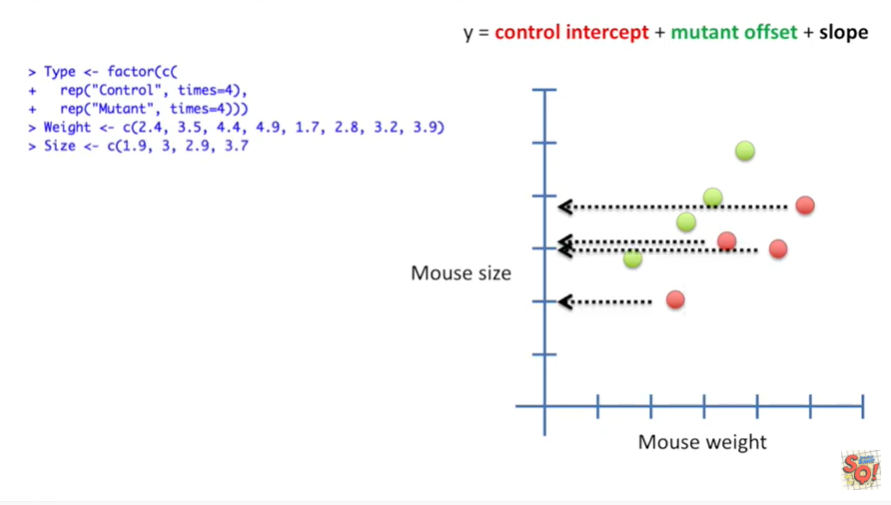
Then we create labels for the mutant mice.



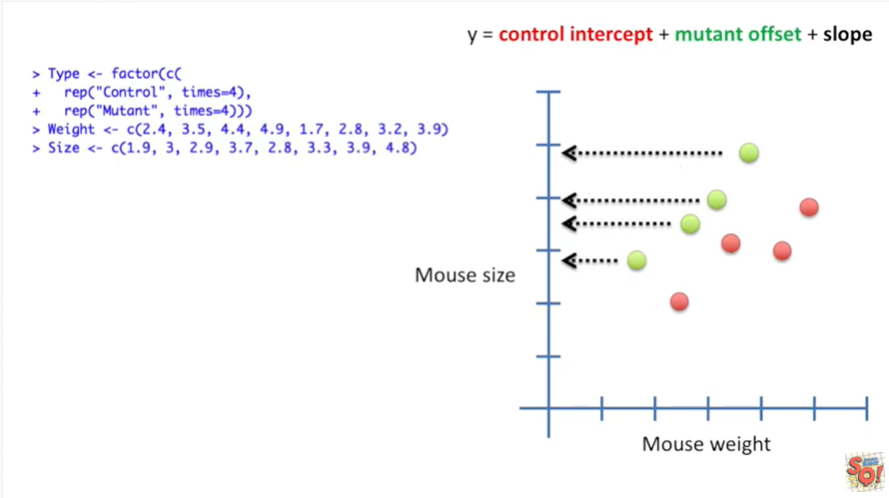
Now we enter the weights for the control mice



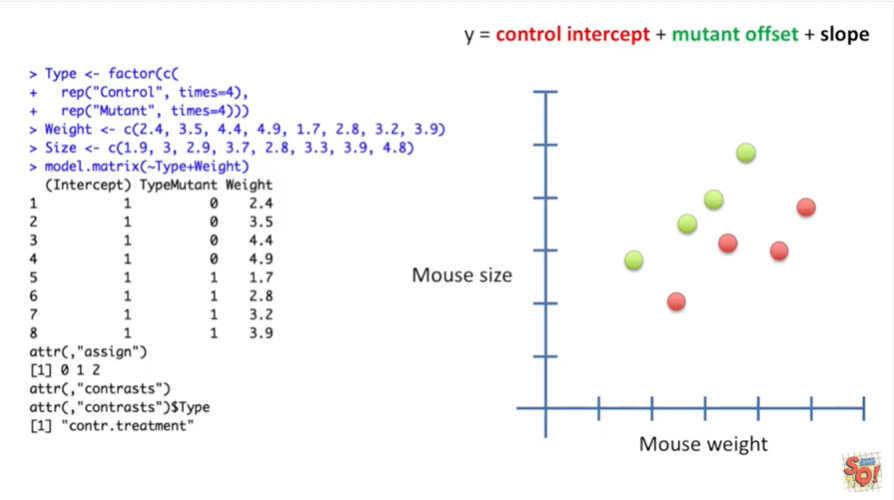
followed by the waits for the mutant mice.



Now we add the size measurements for the control mice



followed by the size measurements for the mutant mice.

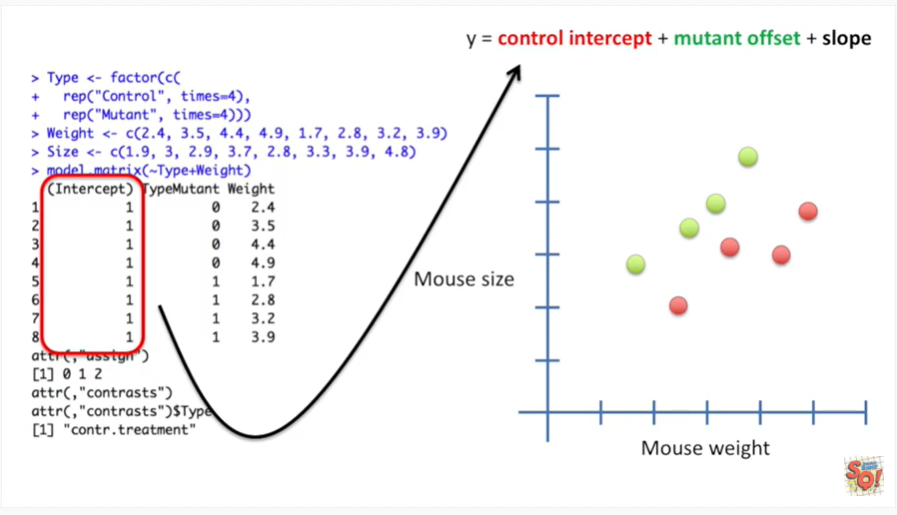


Now we can construct a design matrix.

We do this by telling model dot matrix that we want the Y values or the sizes and we indicate that by a tilde to be modeled by the type of mouse and its weight.

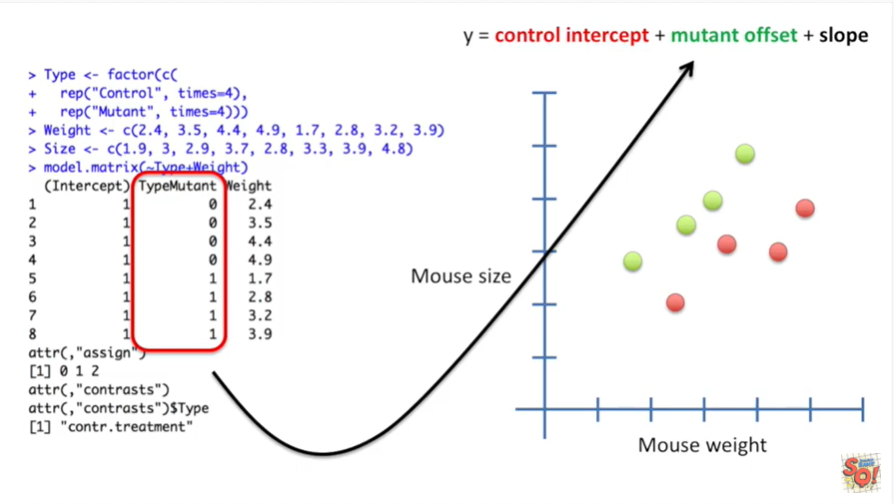
You'll notice that we didn't have to specify that we wanted to term for the intercept our assumes this by default.

Also you don't actually have to call model dot matrix the linear models function will do this for you but I like calling it just to make sure that the design matrix looks the way I expect it to.



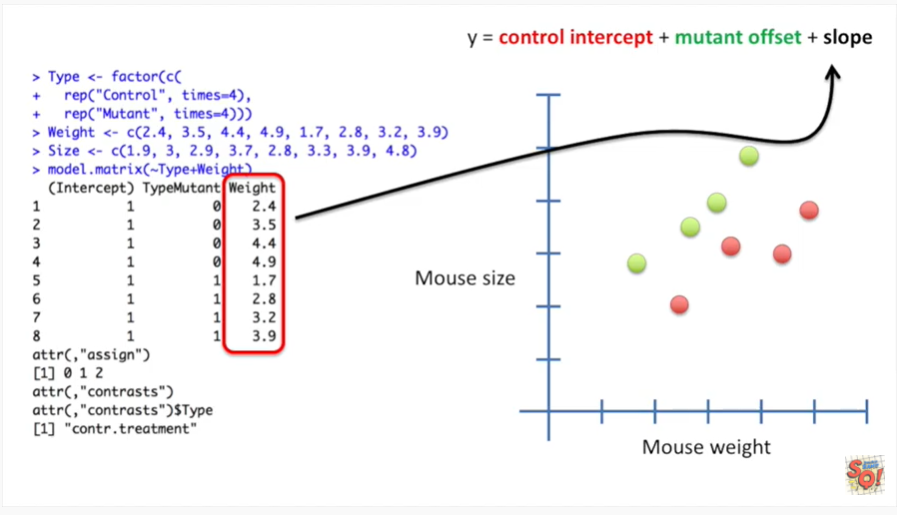
The first column in the design matrix will be multiplied by the control intercept.

In this case the control intercept will be on for all of the measurements.

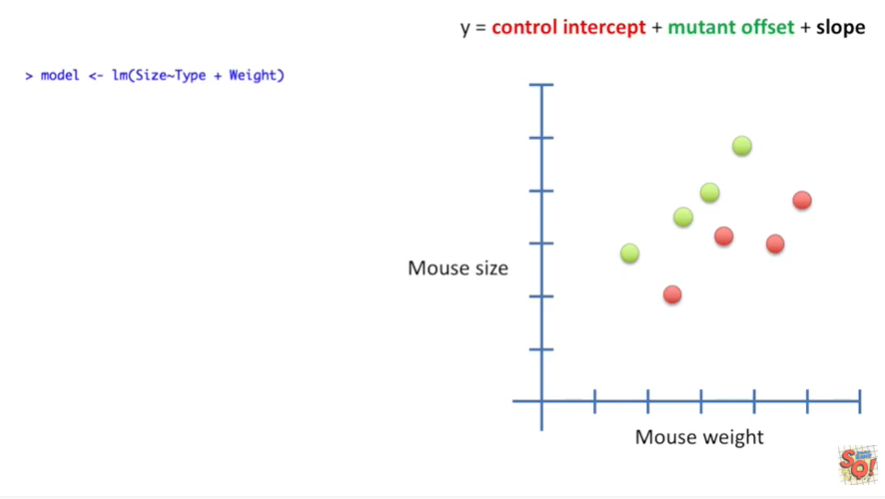


The second column in the design matrix will be multiplied by the mutant offset.

In this case only the mutant values will turn the mutant offset on.



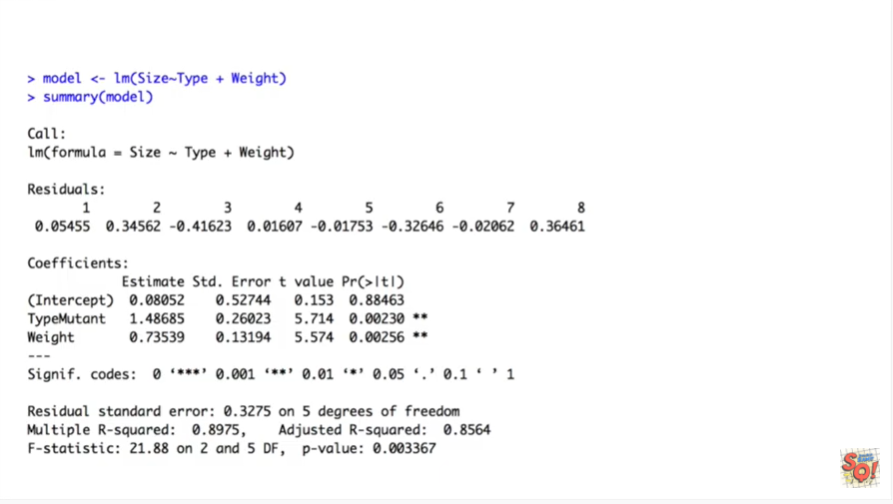
The last column in the design matrix will be multiplied by the slope.



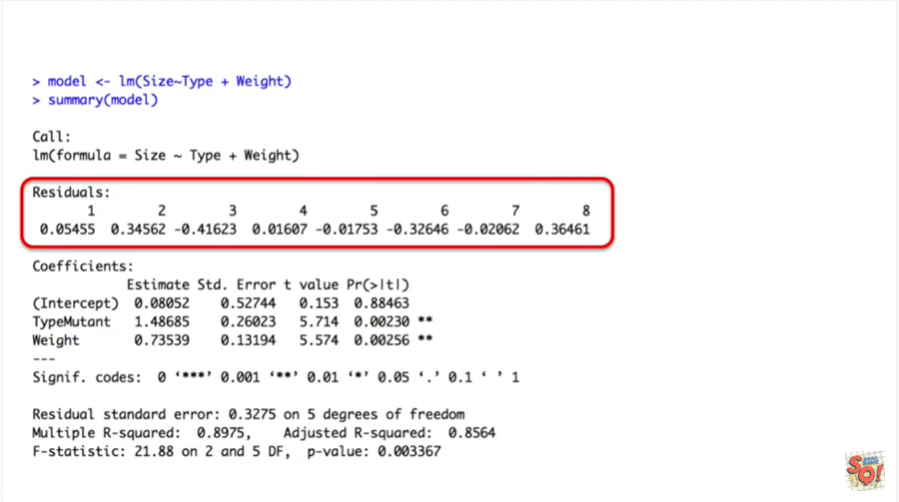
We then call the LM function.

L M stands for linear models.

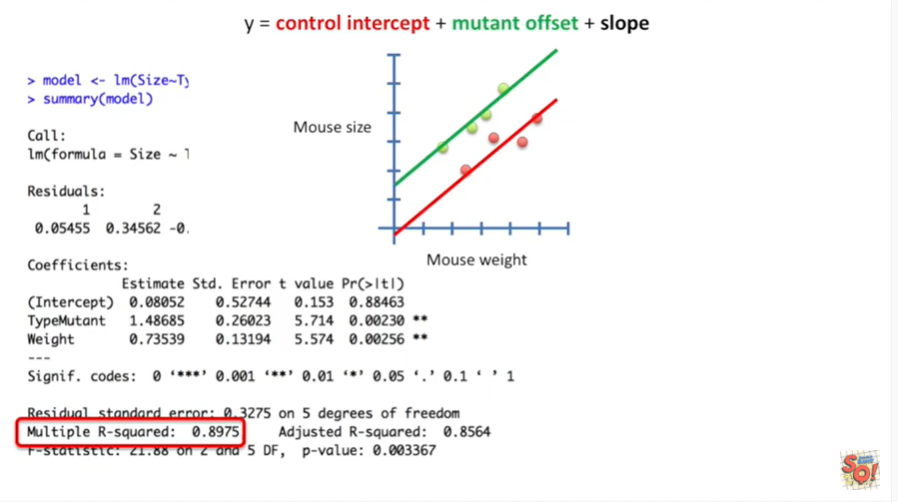
It'll do the least squares fit and calculate the statistics for us.



And here's the summary of all that stuff that the LM function did for us.

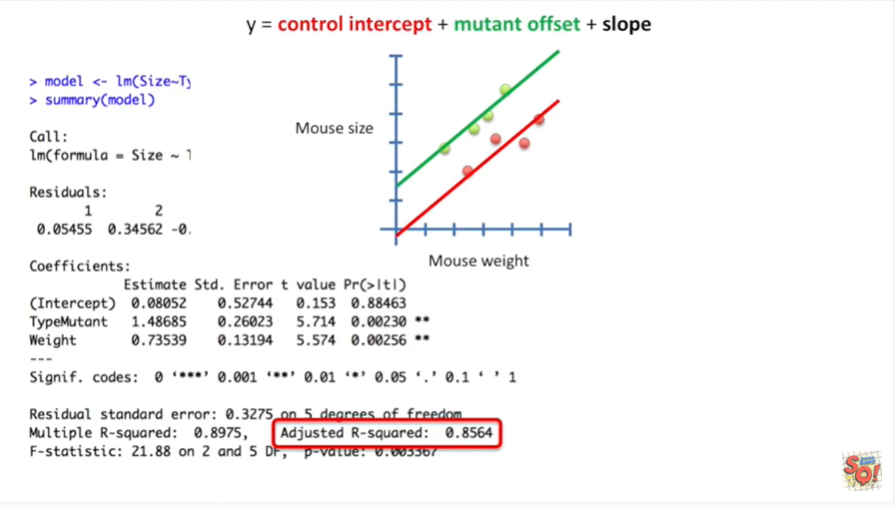


The first thing it does is the least squares fit and then it calculates the residuals.

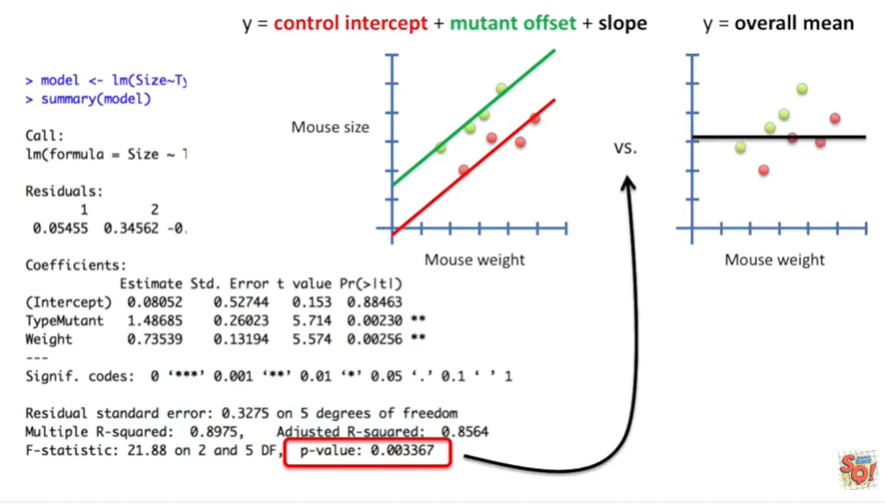


It also calculates the r-squared value for us.

This is called the multiple r-squared because we're fitting a rather complicated equation to the data.

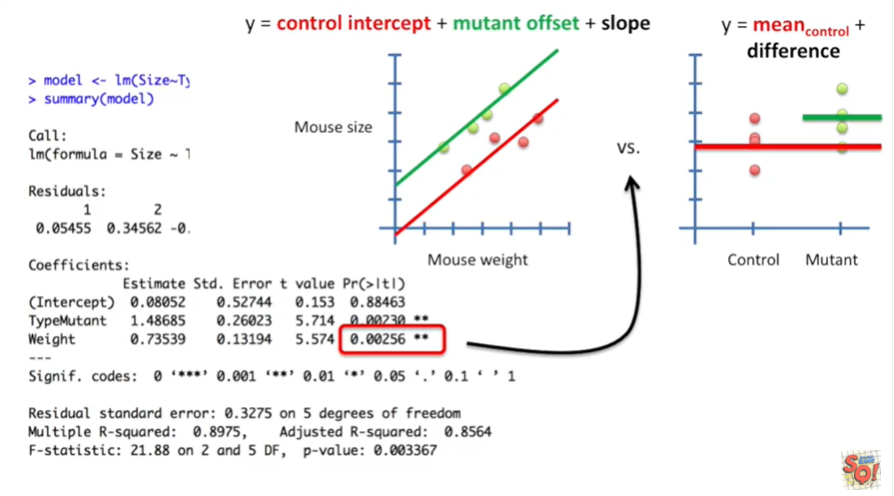


The adjusted r-squared is the R squared value adjusted for the number of parameters in the equation that we fit to the data.



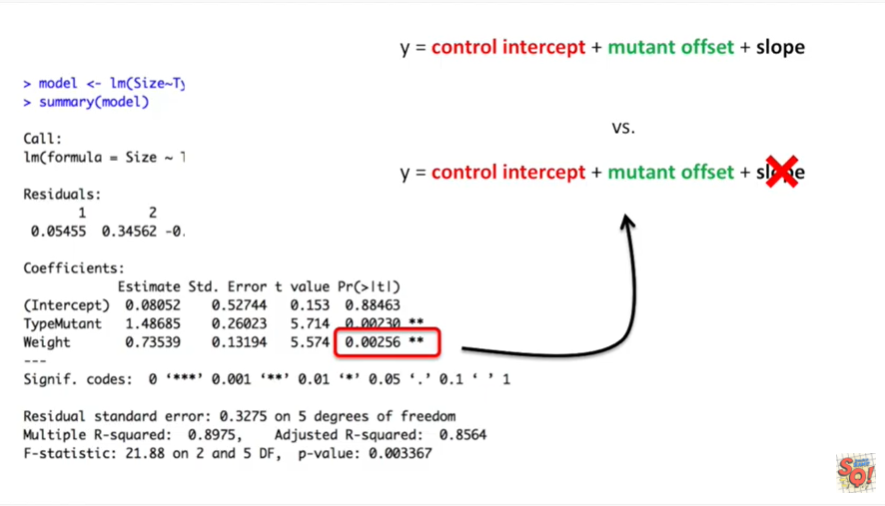
Lastly we have a p-value.

This p-value compares the fit of our fancy equation to the simplest equation possible y equals the overall mean.

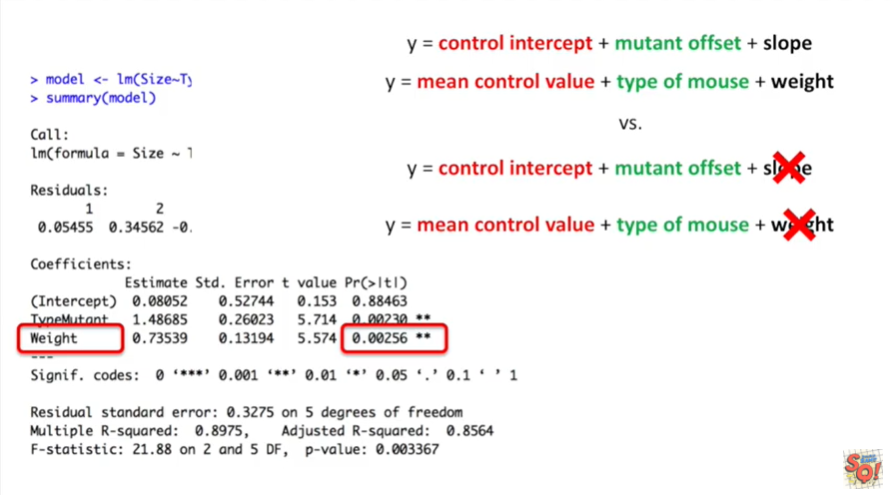


This p-value for the weight parameter tells you how much better the fancy equation fits the data than if we had removed the weight parameter.

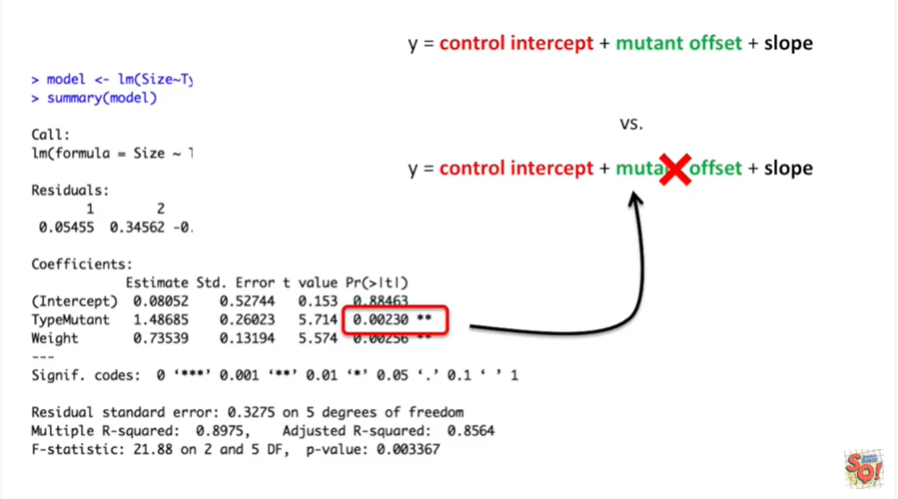
This amounts to comparing the fit of the fancy equation to just a normal t-test.



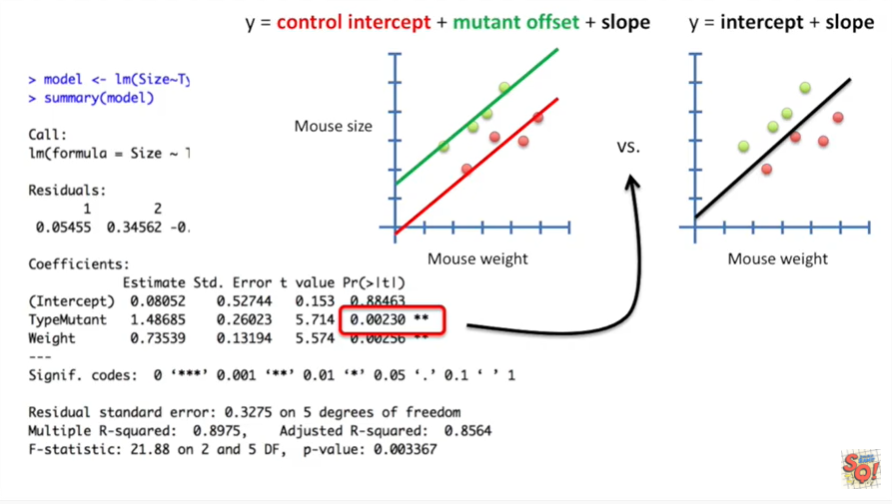
In other words this p-value reflects what would happen if we compared the least squares fit around the original fancy equation to the least squares fit around the fancy equation minus the slope.



And since the slope is the weight this p-value tells us how important the term for the weight is in the fancy equation.



This p-value for the type of mouse tells us how much better the fancy equation fits the data then if we had removed the type of mouse information from the equation.

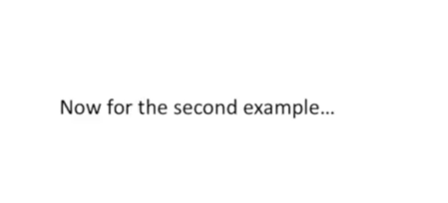


This amounts to comparing the fit of the fancy equation to just a simple linear regression.

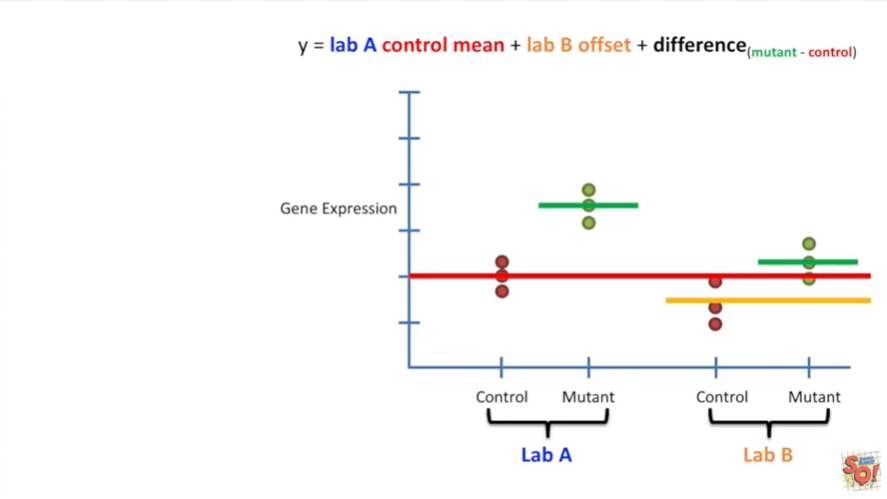
So calling the linear model function actually does a bunch a different test for us.

It's important to know which test were actually interested.

In the next example shows that sometimes were not actually interested in the p value in the lower right hand corner.



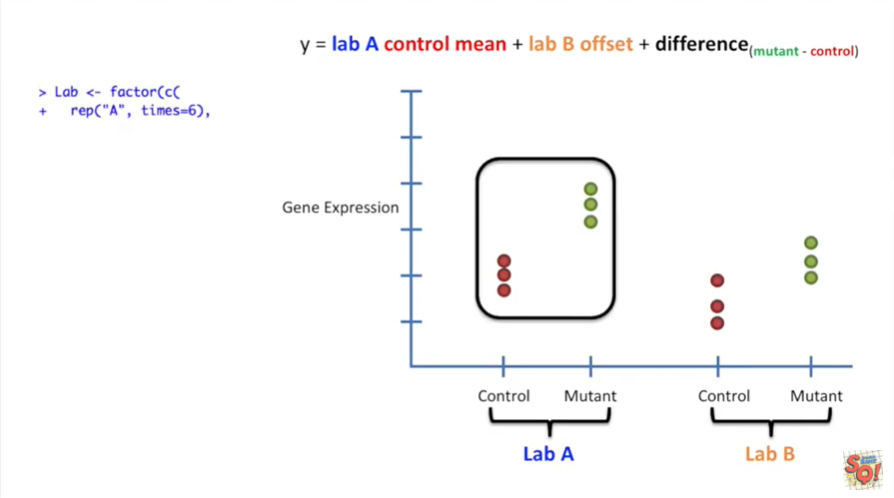
So now let's move on to the second example.



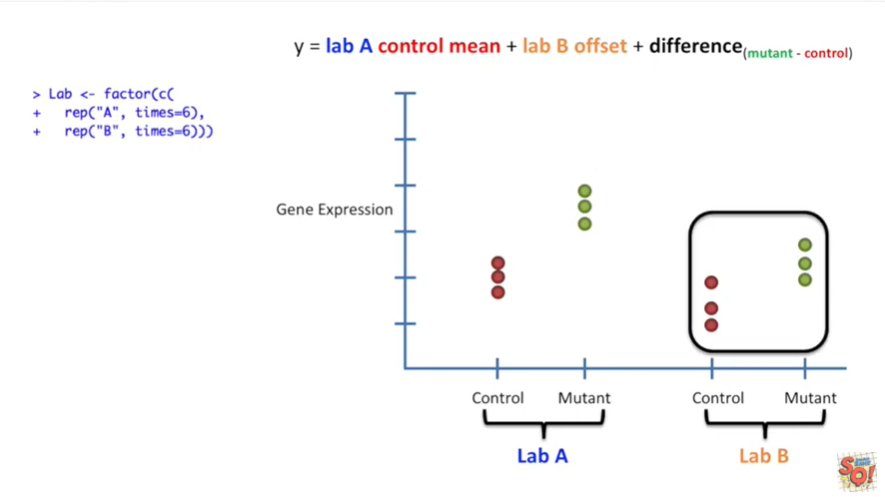
In this second example we compared experiments done by two different labs.

There was a batch effect.

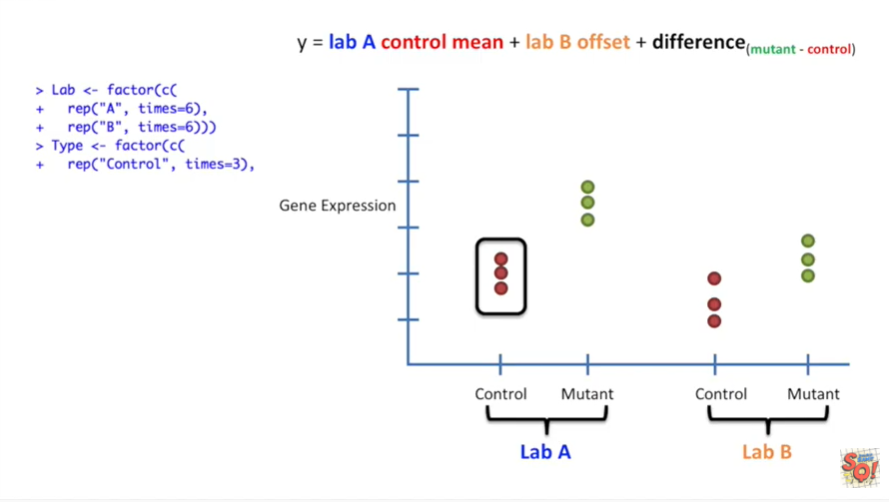
And we wanted to control for that.



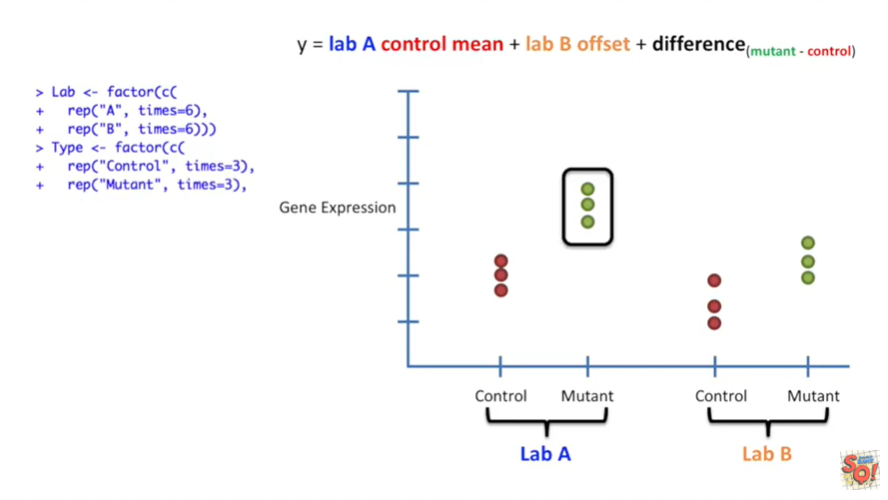
In R we start by creating the labels for the data that was generated by lab a.



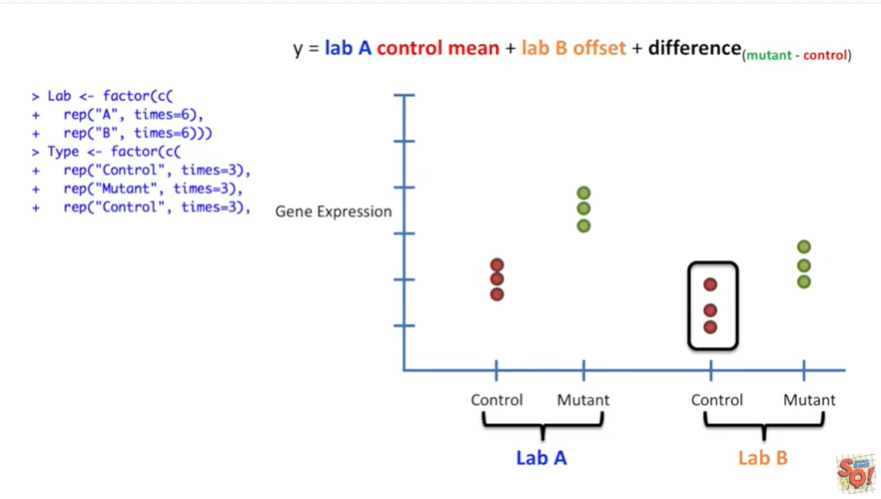
Then we create the labels for the data that was generated by lab B .



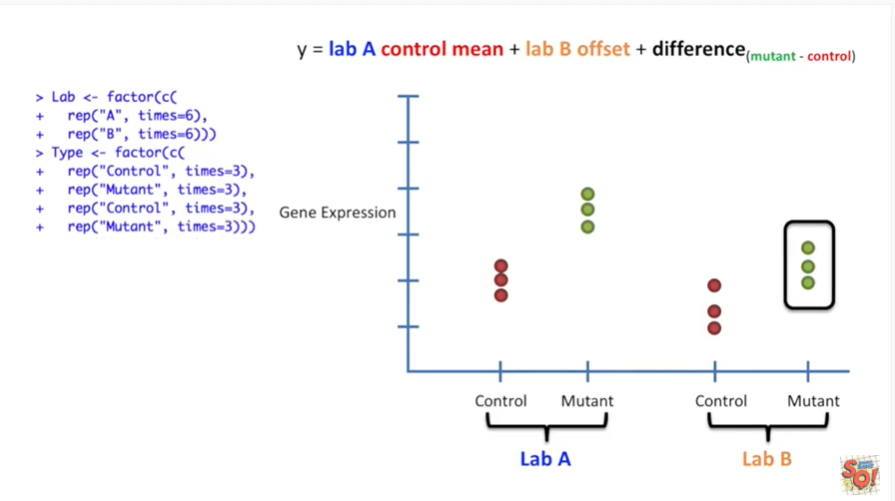
Then we create the labels for the control data from lab a.



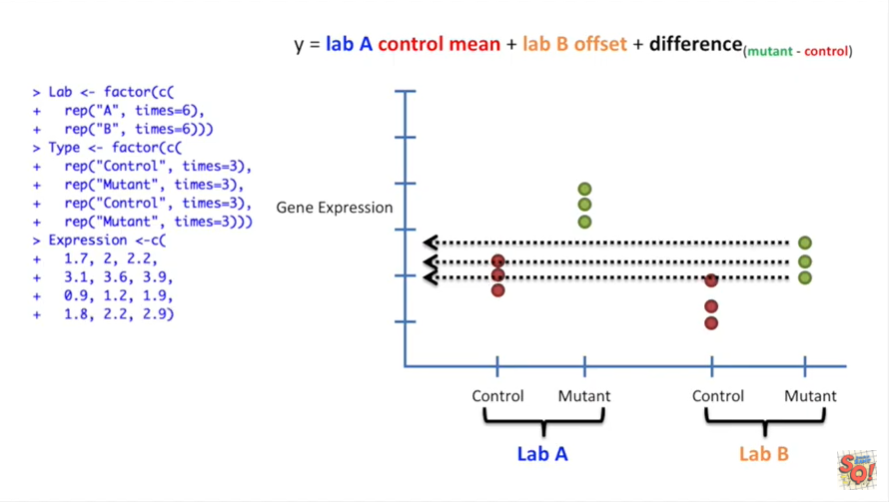
And the mutant data from lab a.



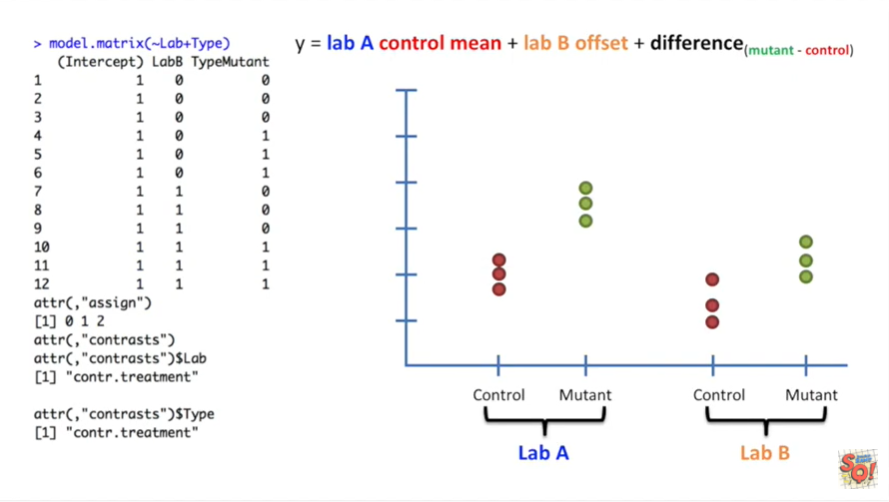
Then we create the labels for the control data from lab B.



And the mutant data from lab B.



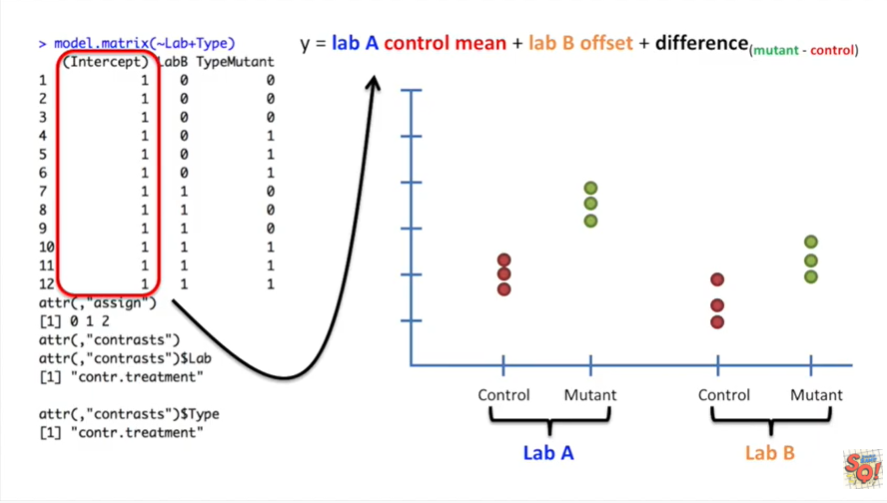
Then we enter in all of the expression values.



Then if we wanted to we could call model dot matrix to see what the design matrix looked like.

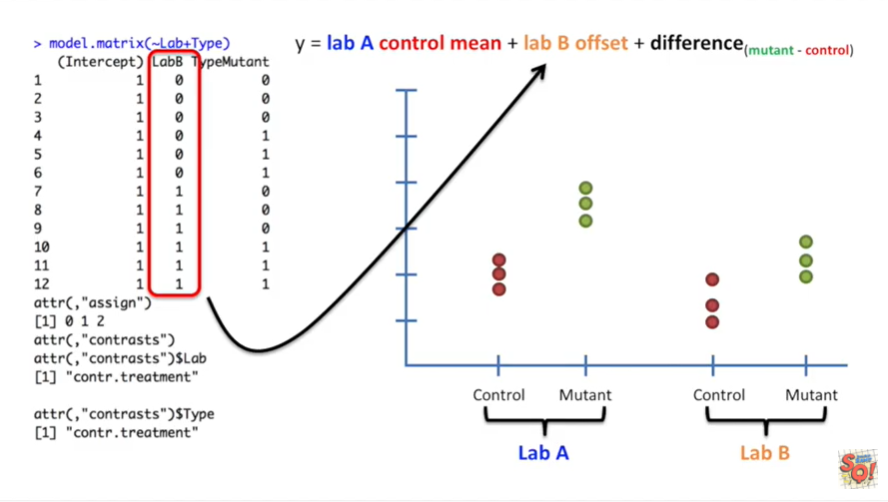
To do this we enter the tilde which represents the Y value or gene expression data and say that it is modeled by the lab and the type.

We don't have to specify the control mean because that is done by default.



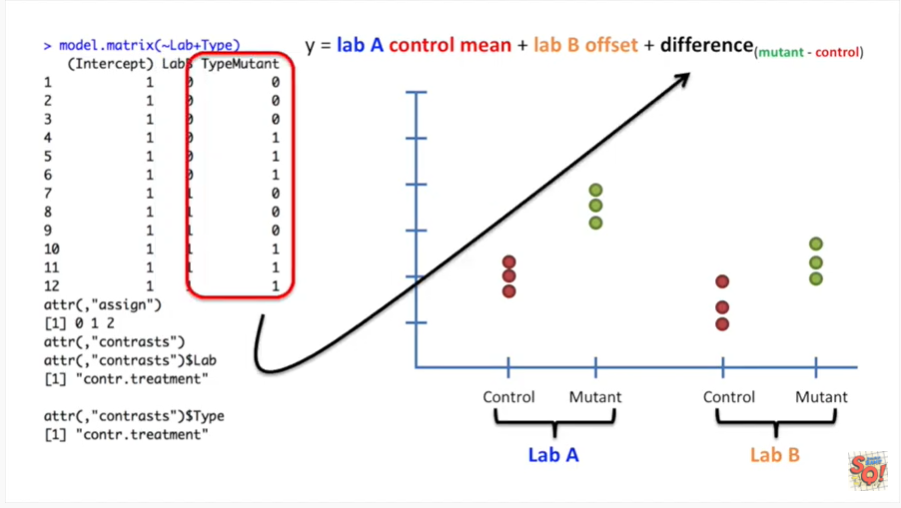
The first column in the design matrix is multiplied by the term for the lab a control mean.

This means that this value is on for every single measurement in the data set.



The second column is multiplied by the lab B offset.

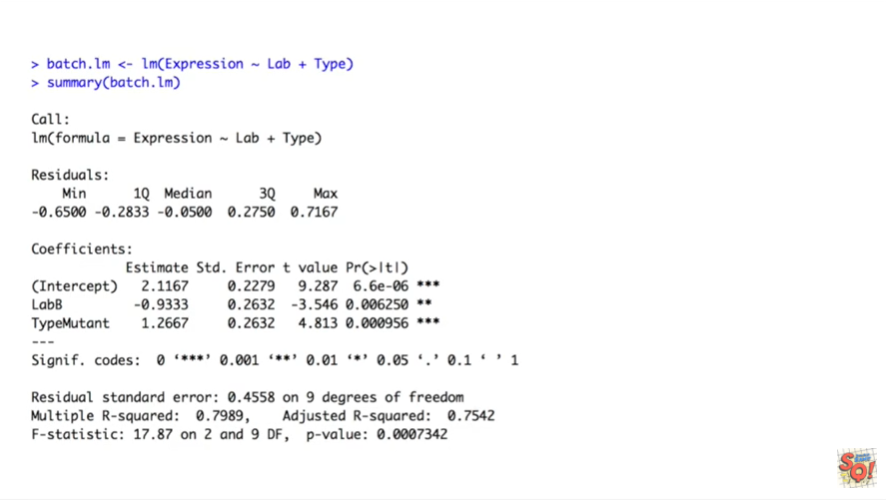
That means that this offset is only on for the data generated by lab B.



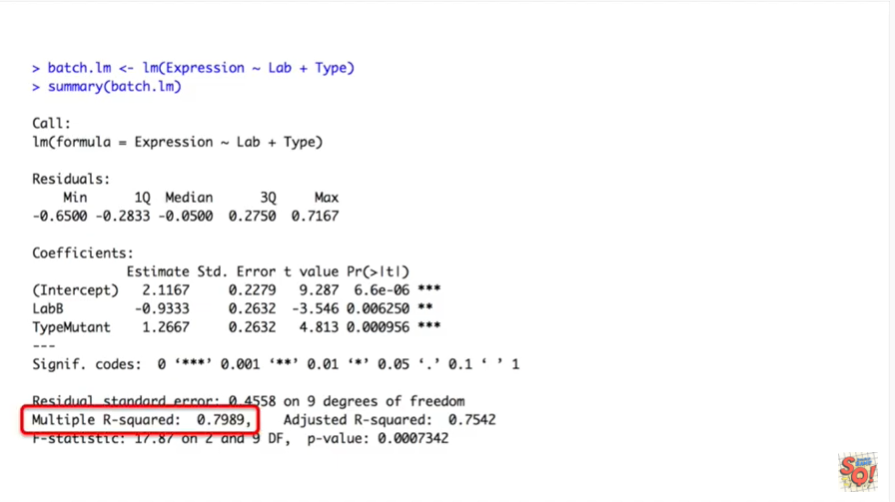
This last column is multiplied by the term for the difference between the mean of the mutant data in the mean of the control data .

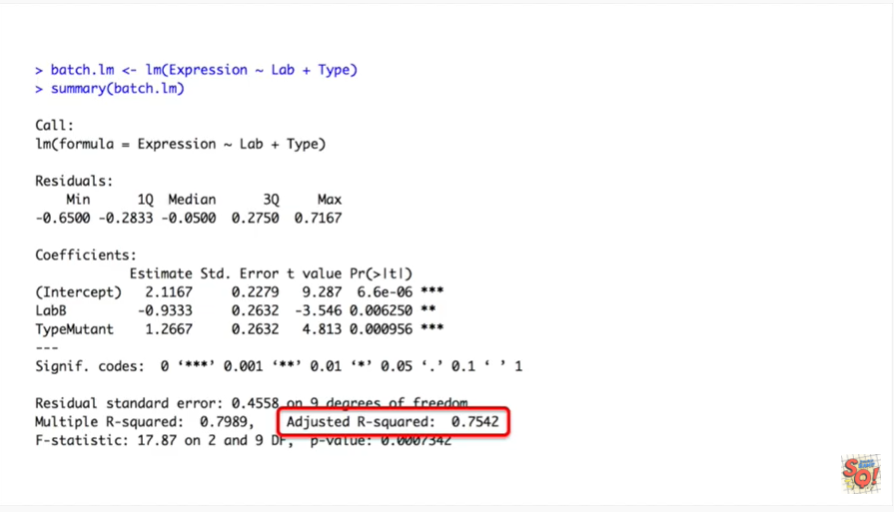


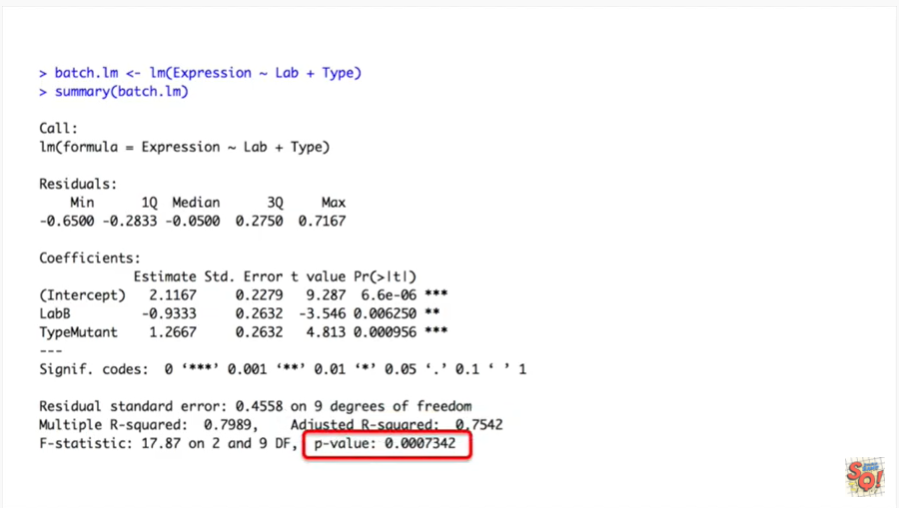
Here's our call to the linear models function.



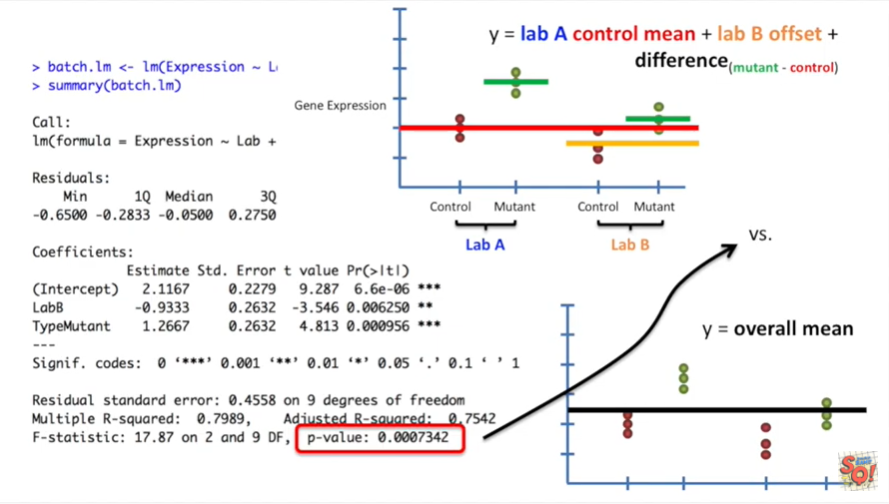
And here's the summary of everything that the linear model function did for us.







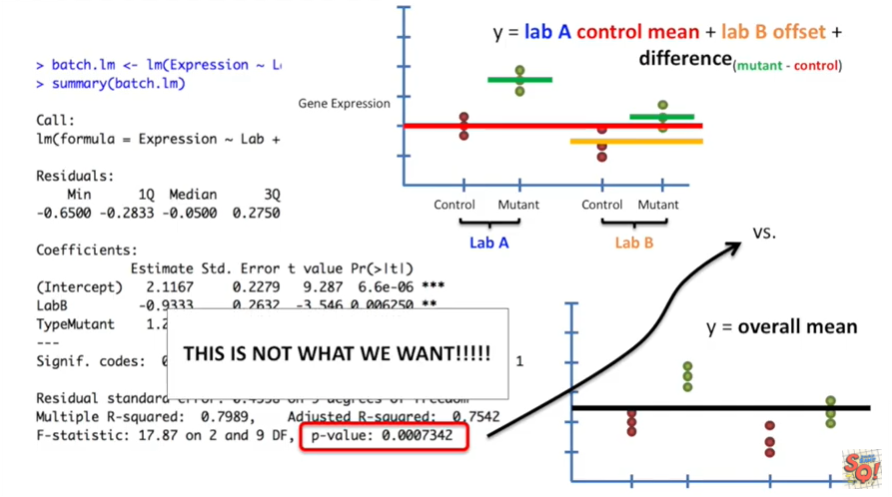
Just like before it has the multiple r-squared and the adjusted r-squared values and then a p-value in the lower right hand corner.



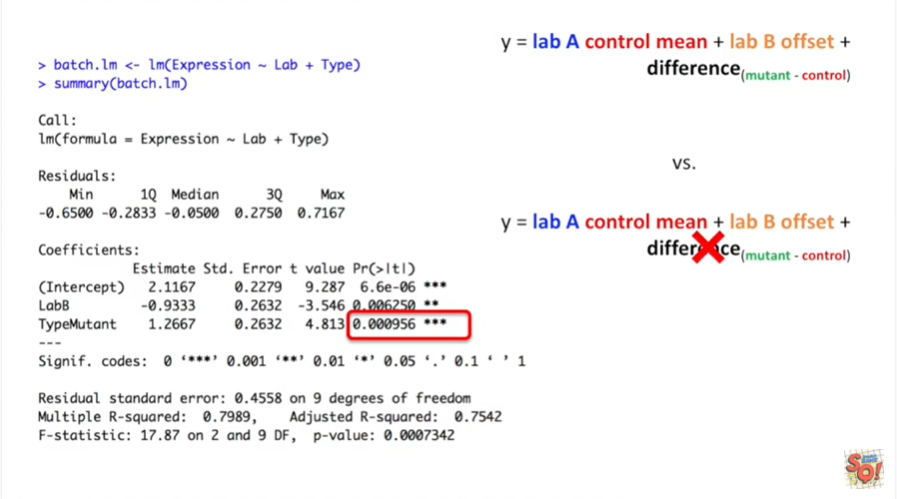
However this is not what we need.

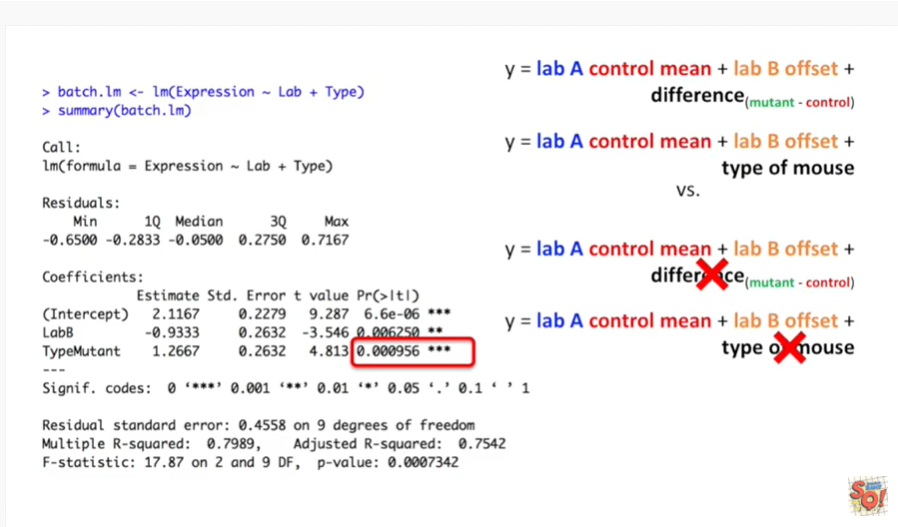
This p-value represents how much better our fancy equation fits the data compared to an equation that is far too simple.

This equation is just y equals the overall mean and it does not test if there is a difference between control and mutant data.

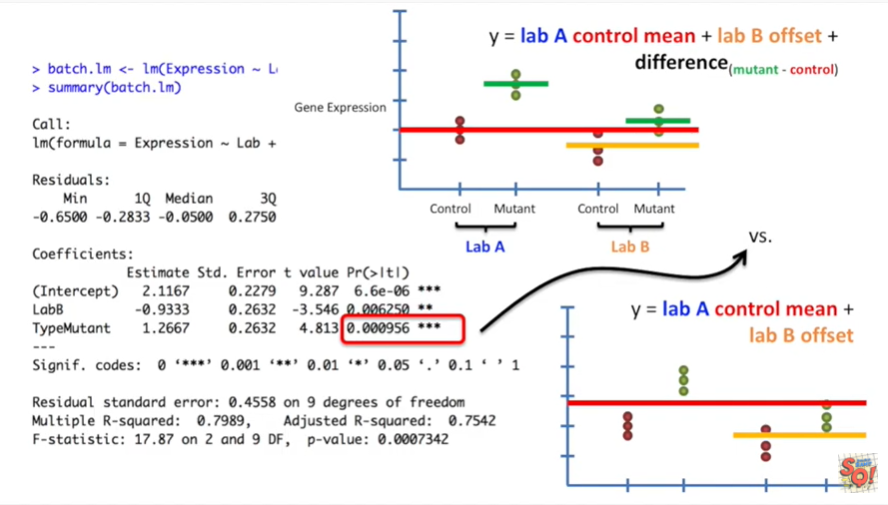


This is not what we want.

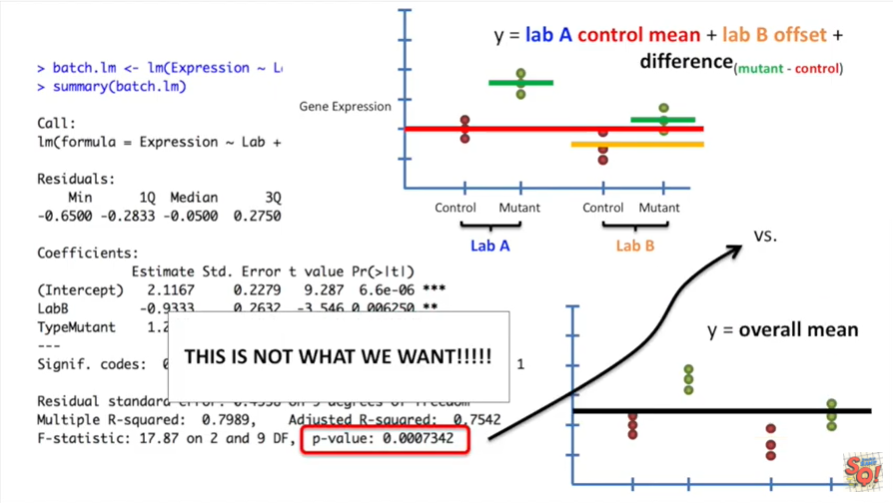




This is what we want or we're testing whether or not we can subtract the term for the different types of mice from our fancy equation and still get a good fit.

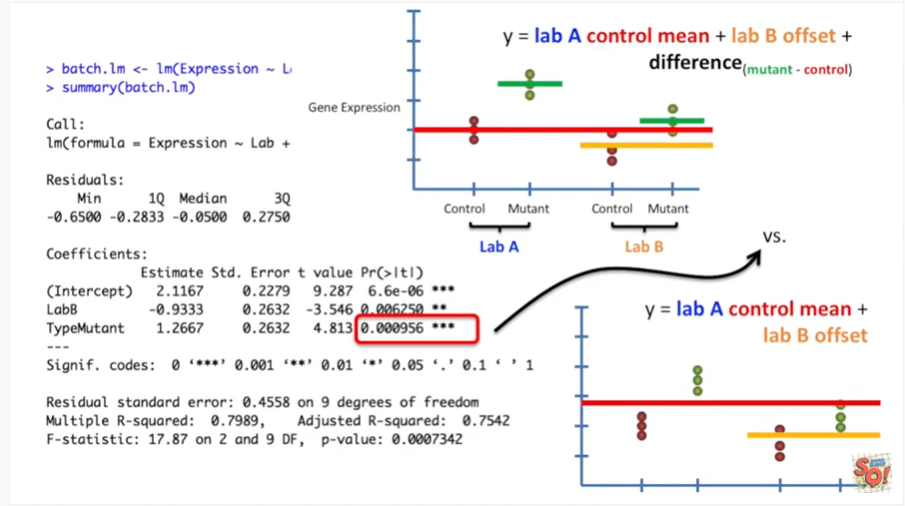


Here's the p value that we really want.



So the moral of the story is that the p value in the bottom right hand corner isn't always the p value that you want.

It reflects the comparison of your fancy equation to the simplest equation possible.



If that's not what you want then you're going to have to look around at the other p values that the linear models function reports maybe that's the one you need.