Week 2

**Models**

Y – Response / Target / Outcome X – Predictor

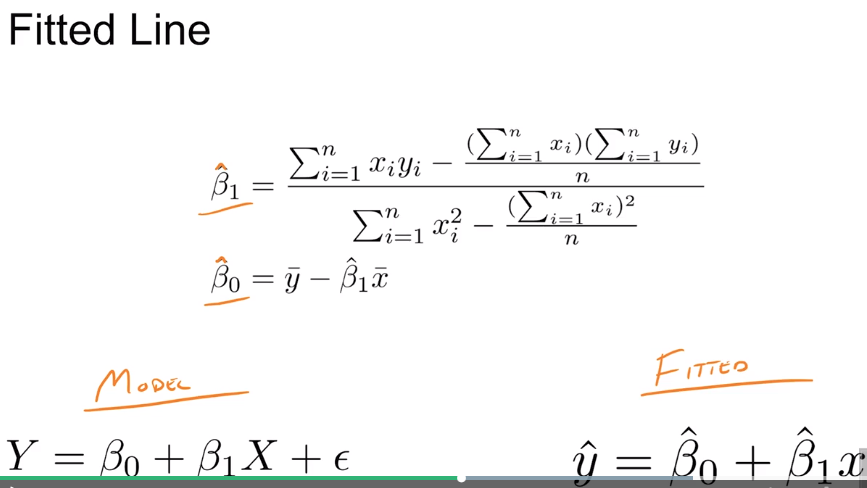
Model to approximate when we drive a car at a certain speed, how long will it take to stop.

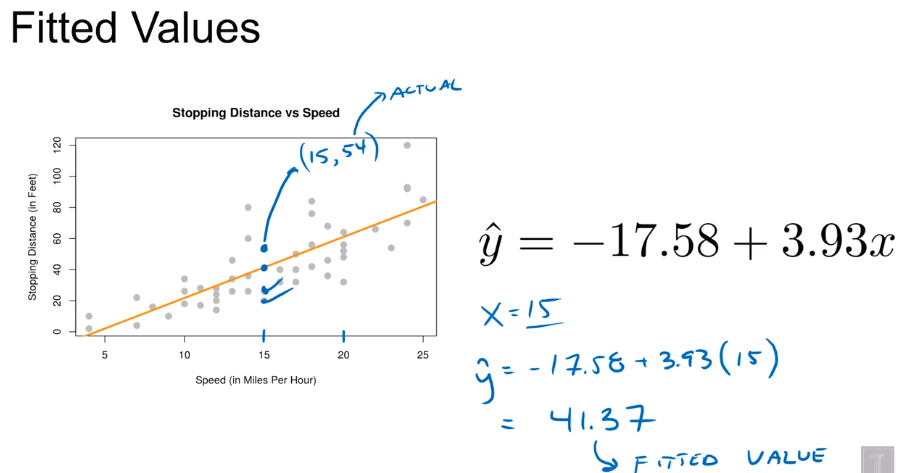
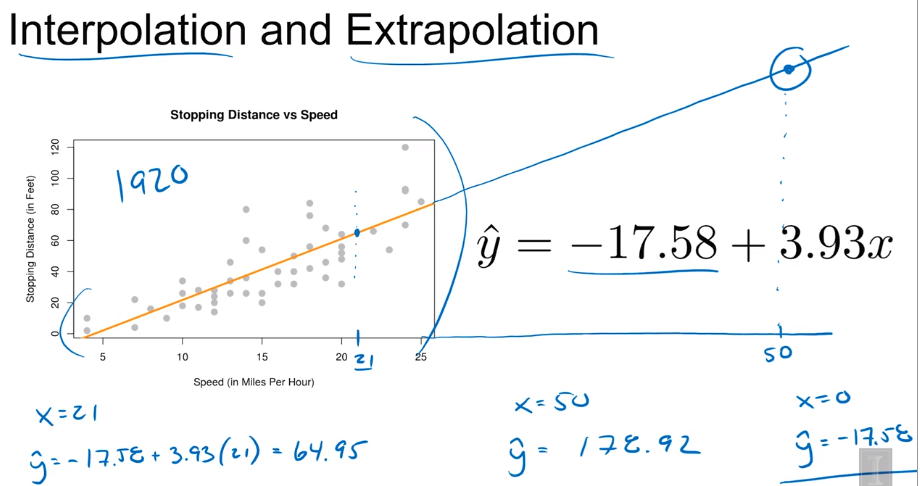
We want to say that the response, Y, is some function of the predictor variable, but we also want to allow for some error. Signal/trend + Noise. Explained + Unexplained. Speed is explaining some of the relationship with stopping distance but there is more going on here because there could be different cars, which explain the different stopping distances. Maybe these cars were being driven on different road conditions, or maybe we're simply measuring poorly

A model being sort of an approximation to reality, do we think that the truth is this exact linear relationship between speed and stopping distance? Probably not. But it will serve a purpose. It will provide a model for the situation. We could say then make predictions like this and we can sort of use it to generally explain the relationship between speed and stopping distance. While the previous two models, the overfitting and underfitting model, are just clearly wrong for this situation, the linear model is sort of a good approximation to reality.

**Fitting a Line**

Once we decide to fit a linear Model assuming the signal is a linear function of x, what is the line that best summarizes this data? So what we want, is for the errors that this line is making, to be small. So a beta 0 and a beta 1, it gives us a line with small error. So the first here, we could find the slope and intercept that gives us the smallest maximum error.

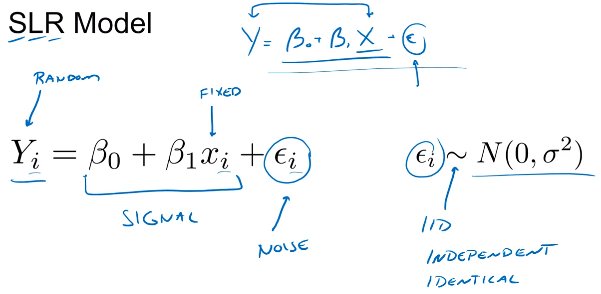
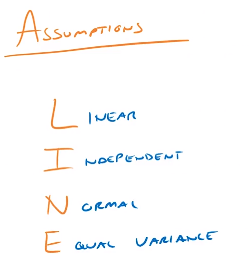
 

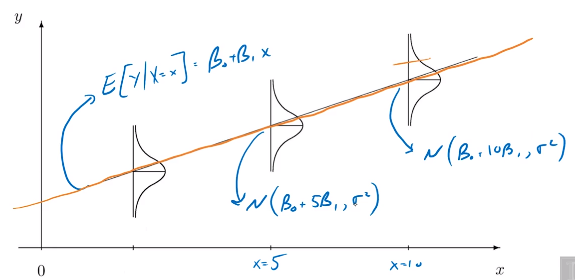
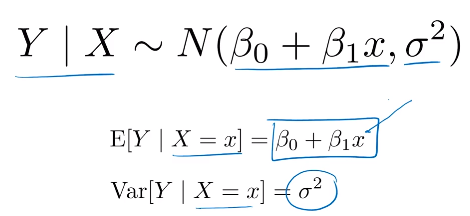
**Simple Linear Regression Model**

generic model, where Y is a linear function of X plus some noise. we'd like noise to be sort of random about this line. So now we'd like to talk about the Simple Linear Regression model. The S for simple has to do with the fact that we only have one predictive variable X. Linear because we have this linear function of X here, and regression because we're looking at the relationship between two variables.

Now, notice that Y is capital, while x is lowercased. And that's to say that in this model, we're assuming that the xi are fixed known quantities. But now, because we're going to introduce a distributional aspect to the noise, the Yis are going to be random. This is because we're now going to assume that the epsilon i's follow a normal distribution with mean zero and variance Sigma squared. And in particular we would say that they are IID, the first i for independent, the second i for identically distributed or same distribution.

We now have a probability model instead of this more generic model after additional information about the noise. So Y has a conditional distribution given a value of X. we're not going to consider x random. For different values of X, we have a different mean, or signal portion. But, for any value of X, we have the same variance, or sort of noise component. So again, the expected value of Y for a particular value of X is that linear function, and does depend on x. Whereas, the variance of Y for any x is simply Sigma squared and does not depend on x.

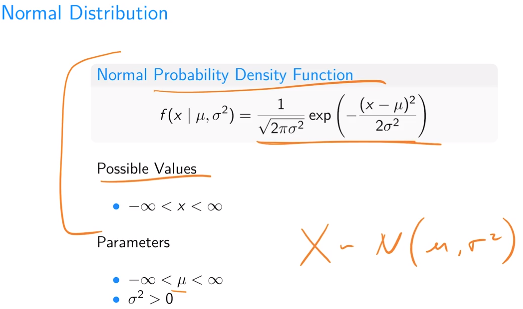
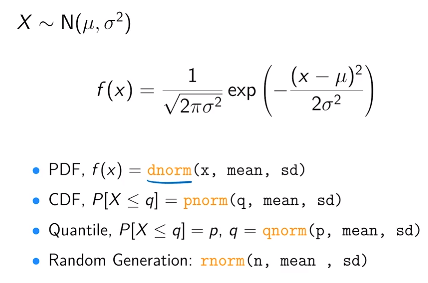


at each value of x there's a different normal distribution, with the mean according to the line, but they all have the same variance. So that's why we see these three normal distributions here, all have the same shape. So essentially, signal component is the mean of all these normal distributions. And then about this signal we expect most of the data to be within one standard deviation according to the normal distribution.

Sometimes we'll want to directly discuss what we call the assumptions of the simple linear regression model. We can do that with LINE. So Linear, because the signal is a linear function of x. Independent, because we are assuming that the errors are independent of each other. So an error at one point does not influence an error at another x value. Normal, because we assume that the errors follow a normal distribution, so that observations will be spread out about the line according to a normal distribution. And we're assuming Equal variance. So at any x value the points vary about the line, the same amount.

**Normal Distribution**

As an example of a continuous random variable, we'll talk about a normal random variable and the normal distribution, which is probably the most popular distribution of them all. Normal distribution is given by both its PDF and its possible values. So here, a normal variable will take values anywhere on the real line, and have this density function controlled by two parameters mean and variance of a normal random variable.

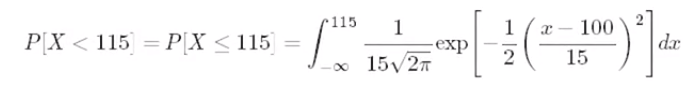
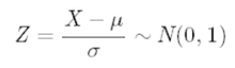
also called the bell curve or the Gaussian distribution. And again, we said that it's controlled by its parameters, mu and sigma squared. So we see that the vast, vast, vast majority of the area under the curve for any normal distribution is within three standard deviations of the mean. sometimes we'll see this referred to as the 68-95-99.7 rule.

So similarly, here are three curves that all have the same mean, but different variances. So this middle curve here is again the standard normal, the lowest curve here, this has a higher variance. And the sort of sharp, tall curve has lower variance. The lower the variance, the tighter all of the area is around the mean, whereas the higher the variance, the more spread out the area is about the mean.

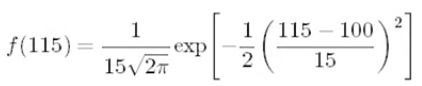
The dnorm function takes a particular value, x, and plugs it into the PDF for us. you just have to remember the standard deviation is the square root of the variance. So there's also the pnorm function, which is the CDF, so it calculates probabilities for us. So it's essentially calculating the integral from negative infinity to a q of that density function. There's also the quantile function called qnorm, if we draw a normal curve here, we input to this function a probabilities for this area to be, let's say 0.15, I want to know the value here, q, that makes that true for a particular normal distribution. And then lastly, there's the rnorm function, which will generate random observations from a particular normal distribution.

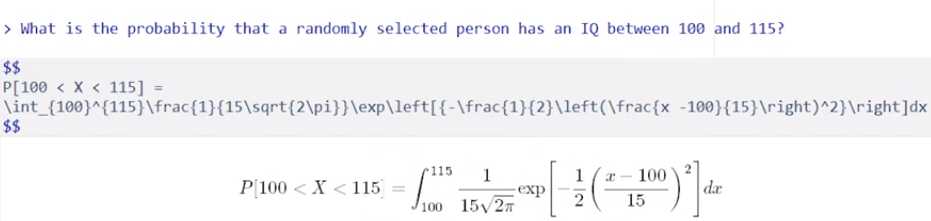
**Normal Distribution in R**

IQ score – Normal distribution with a mean of 100 and SD of 15. X random variable representing a score after randomly selecting a person taking test.

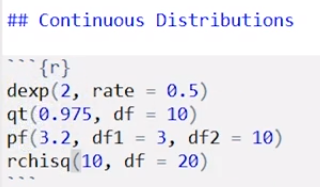
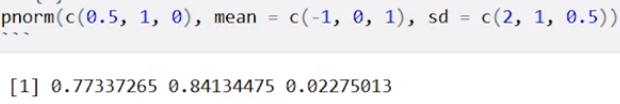
 Pnorm(115, 100, 15) = 0.8413  Z = (115 -100)/15 = 1 SD or pnorm(1) = pnorm(1,0,1) = 0.8413

What is the height of the density curve at IQ = 115? dnorm(115, mean=100, sd=15) = 0.0161. D=Density

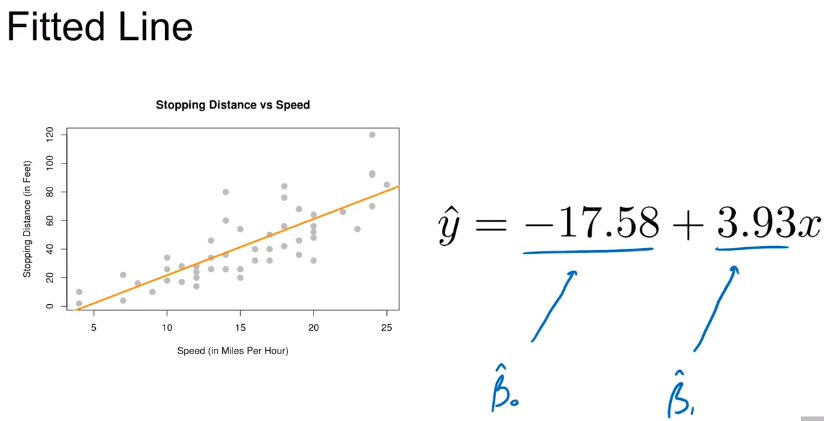
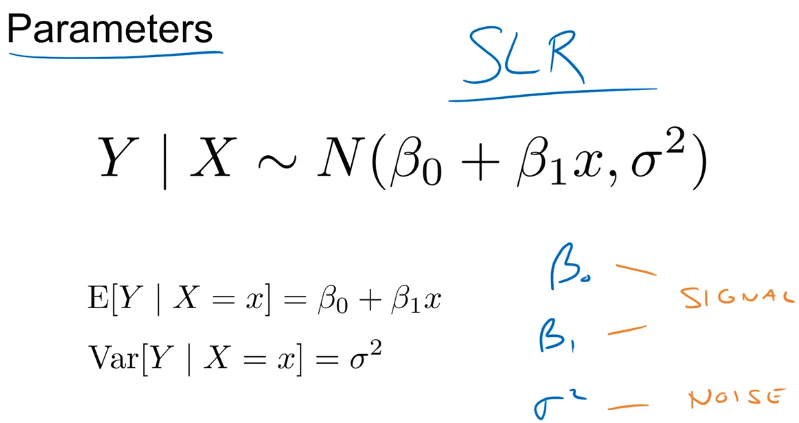




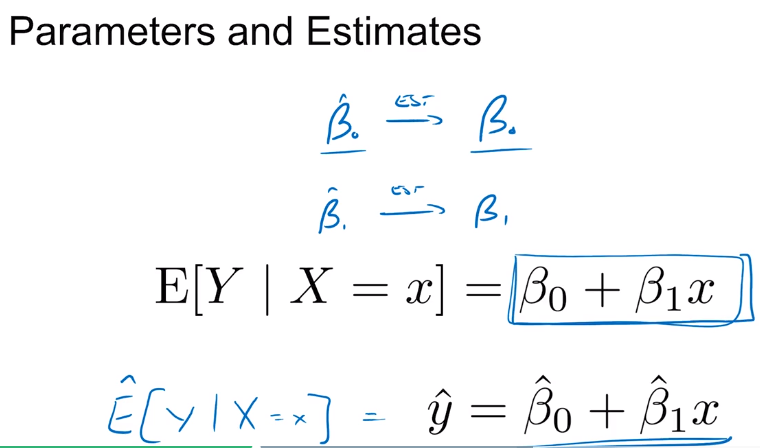
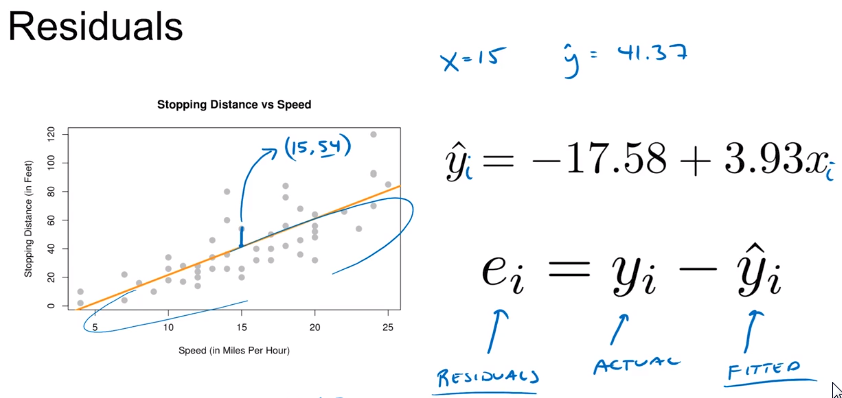
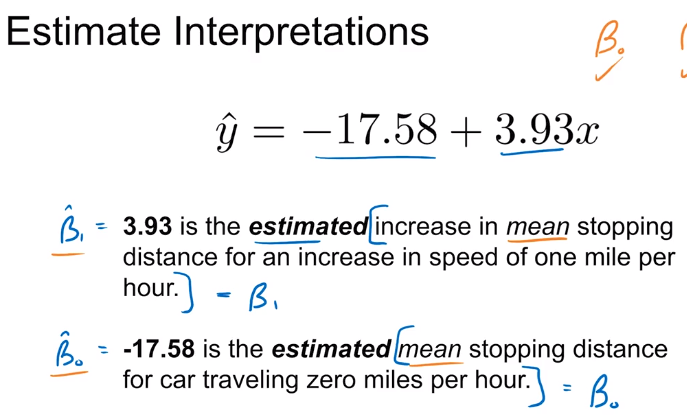
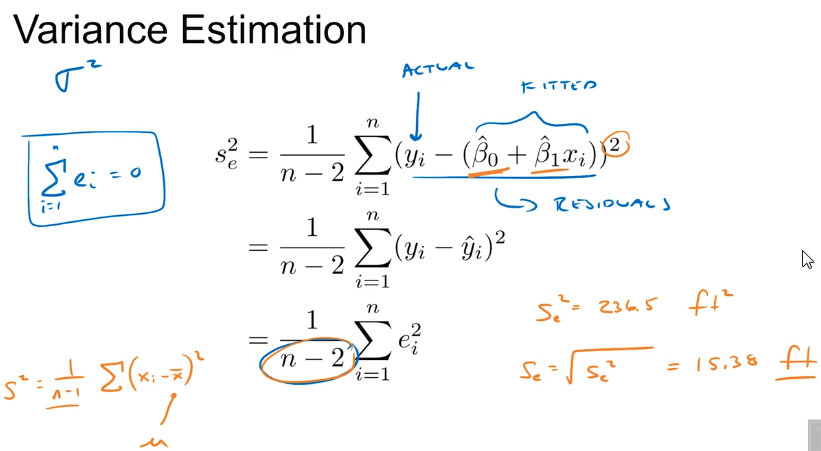
|  |  |
| --- | --- |
|  | pnorm(115, 100, 15) – pnorm(100,100,15) = 0.3413 = diff(pnorm(c(100,115),100,15)) |
|  | pnorm(130,100,15,lower.tail=FALSE) = 1 – pnorm(130,100,15) |
|  | qnorm(0.95,100,15) = 124.67  qnorm(0.05,100,15,lower.tail=FALSE) |
| What is the probability that someone has IQ more than two SD form mean. P[ |X-100| > 30 ] | 1 - diff(pnorm(c(70,130), 100, 15))  2 \* pnorm(70,100,130)  2 \* pnorm(2, lower.tail = FALSE) |

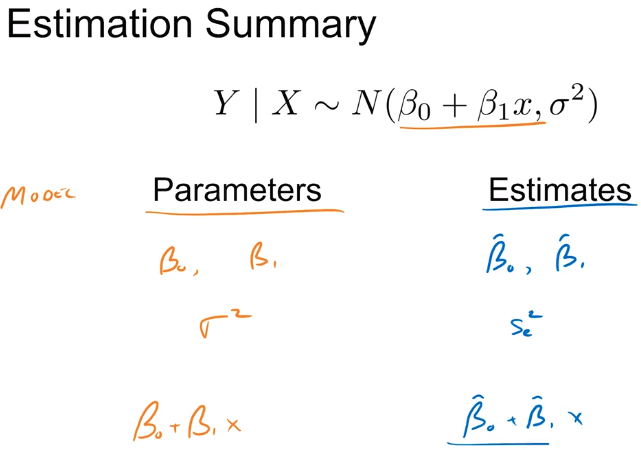
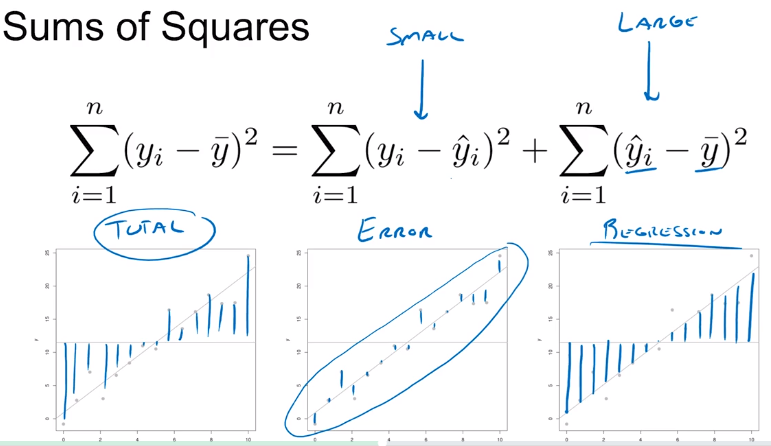
Vectorization 

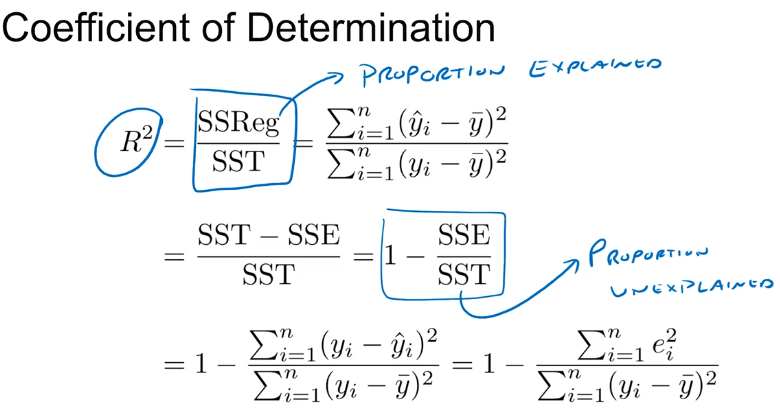
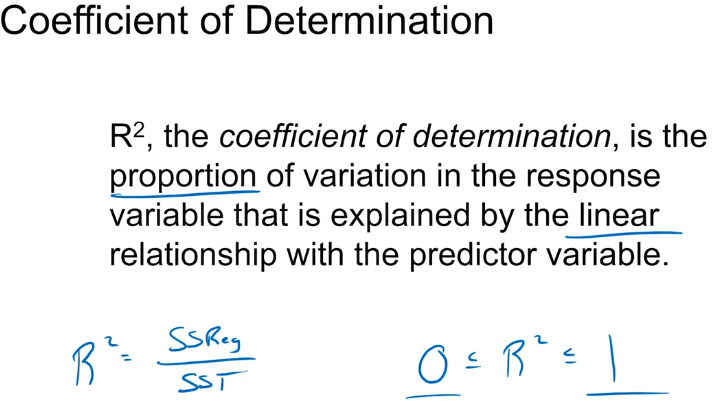
**Estimation**

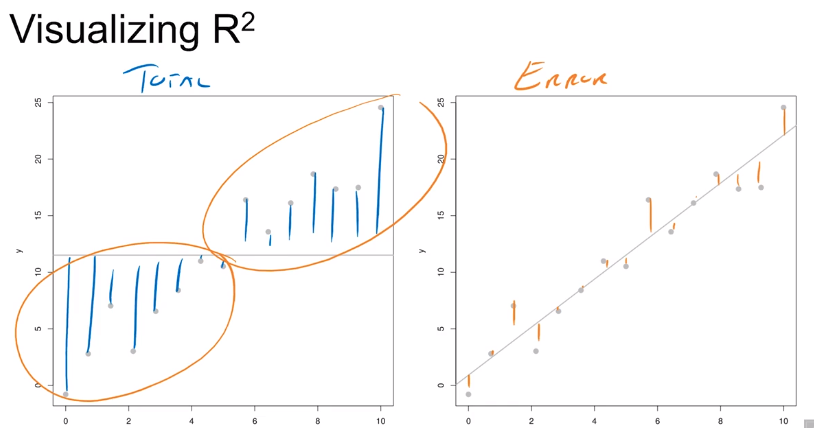
 

model has three parameters, beta 0, beta 1, and sigma squared. Beta 0 and beta 1 being parameters for the signal, and sigma squared being a parameter for the noise. beta 0 hat and beta 1 hat are sample statistics calculated from data and are estimates of the model parameters beta 0 and beta 1.

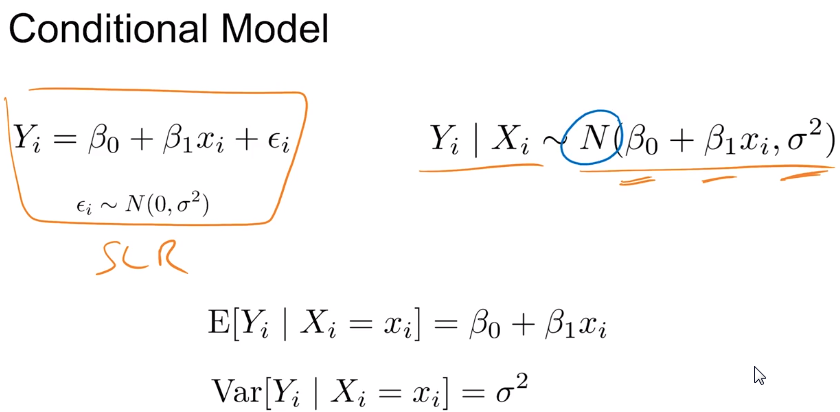
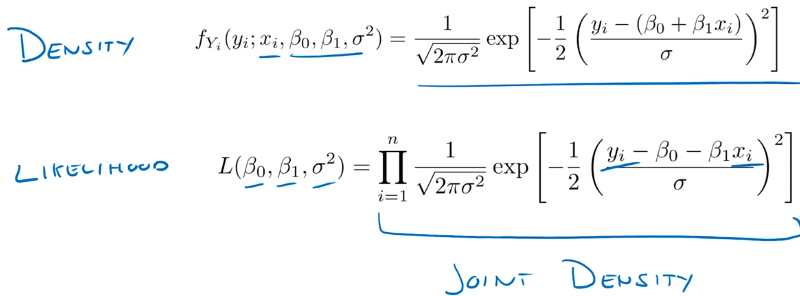
 

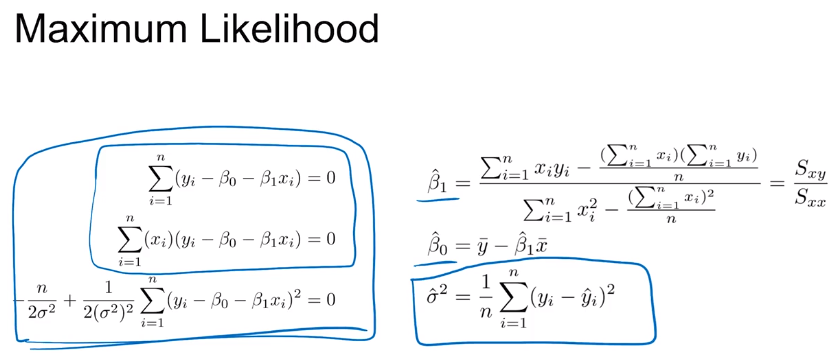
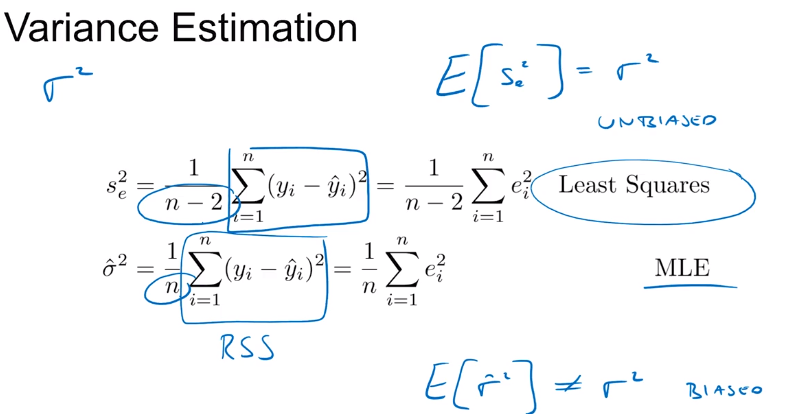
|  |  |
| --- | --- |
|  | if we look at this data set here there's a very clear relationship in this data. It looks something like this but because it's not a linear relationship when we fit loose squares, we probably get a line that looks maybe something like this. And this would result in R square that's very close to 0. So while R squared is 0, so the linear relationship doesn't explain what's going on here, there is still a relationship. |

**Maximum Likelihood**

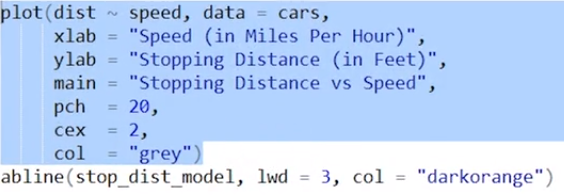
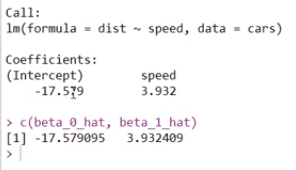
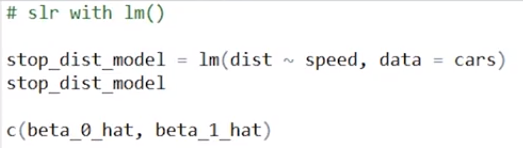
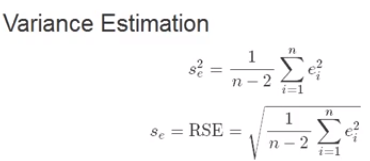
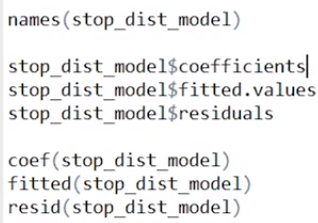
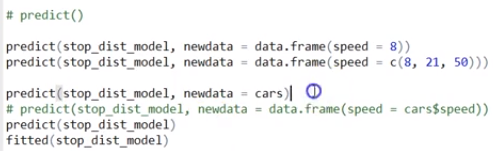
 

We could write this SLR conditionally for a known value of Xi, the Yi's are assumed to have a normal distribution and then we said about estimating beta 0, beta 1 and sigma squared but so far the way we did these was using a method of least squares. So first we essentially found the lines that minimizes the square root errors and after doing so we have beta 0 hat and beta 1 hat which estimated our two model parameters and then we used those to estimate residuals and average that to obtain sigma square estimate. We ustilize the fact that we have a probability model but we dint utlize the fact that y follows a normal distribution, given an X value.

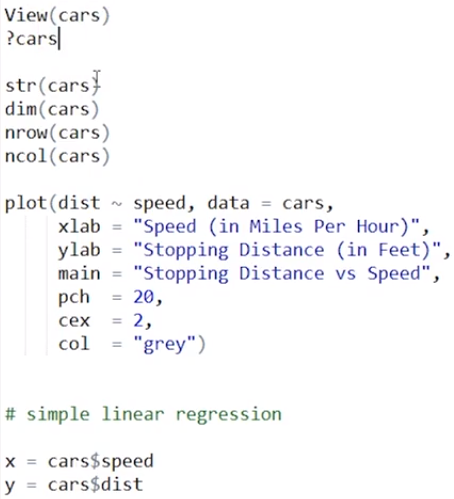
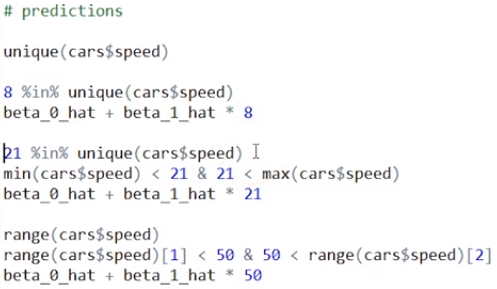
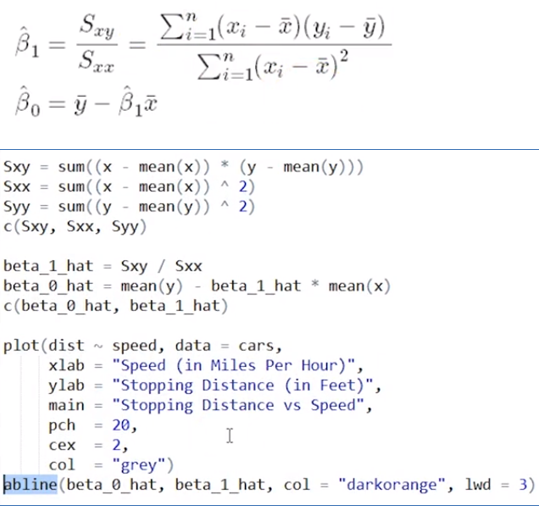
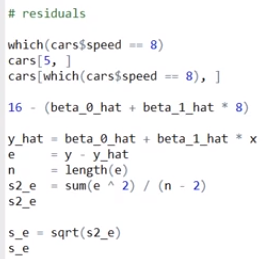
So now we want to look at an alternative way of finding estimates for the three parameters using the fact that we have a probability model. Given the conditional model a value of Xi, we have the density function for each Yi. Now also because we're assuming that the errors are iid, the Yi's are also independent. When we have independent random variables, we can multiply their densities together to obtain the joint density. now we want to go the opposite direction and say, well, we know the data and we want to find the model parameters. So when we're considering data fixed instead of random like we would with densities, we have a likelihood. So, we want to find the values of beta 0, beta 1 and sigma squared that give us a high likelihood of obtaining the data that we saw. So essentially it's the opposite of what we normally do when making probability calculations. Instead of making a probability calculation assuming parameters are known. Now we're calculating a likelihood assuming the data is known. So what we'd actually like to do is maximize this likelihood. So instead of directly maximizing the likelihood, we'll maximize the log likelihood

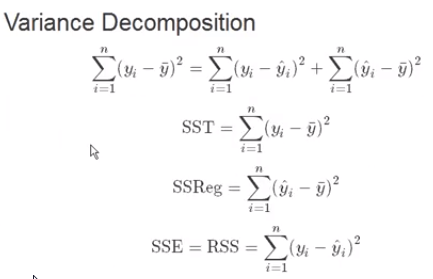
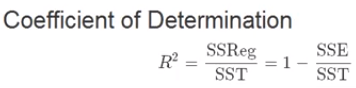
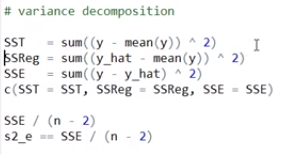
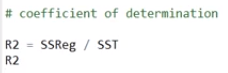
 

**The lm() Function**

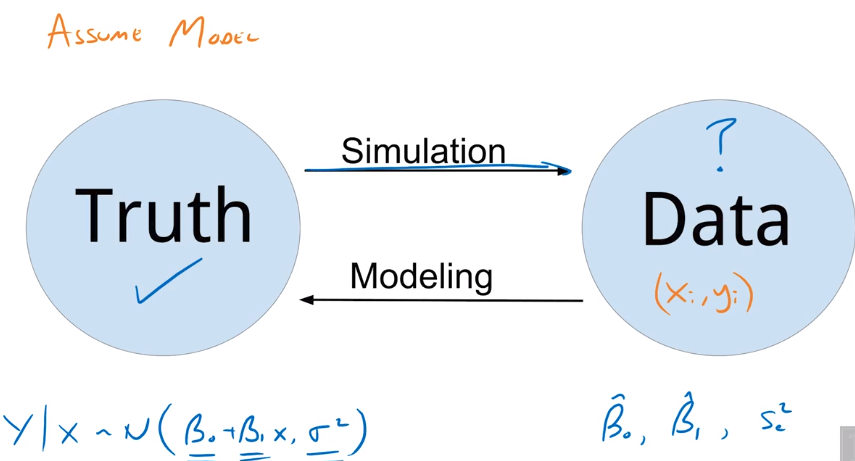
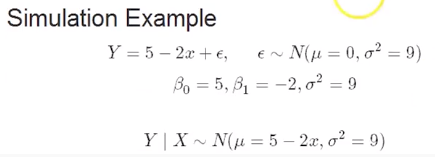
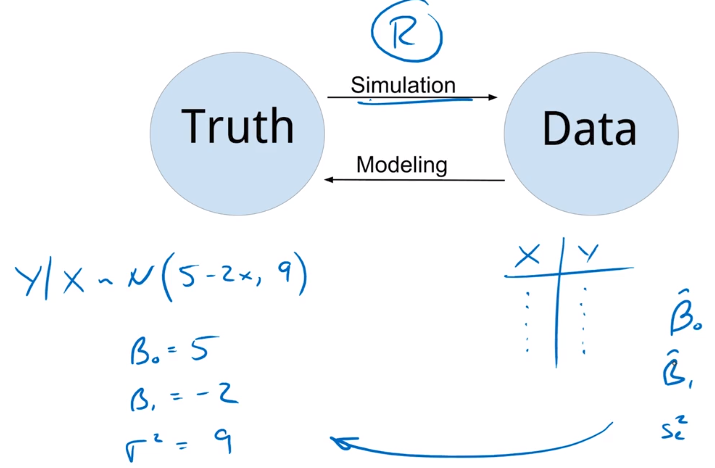
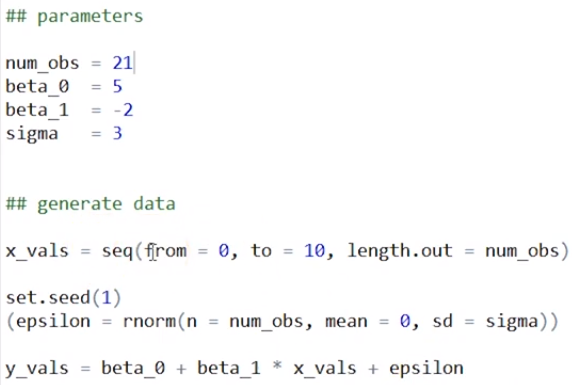
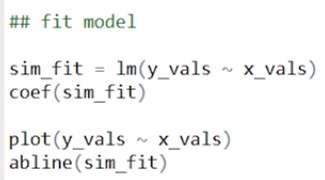
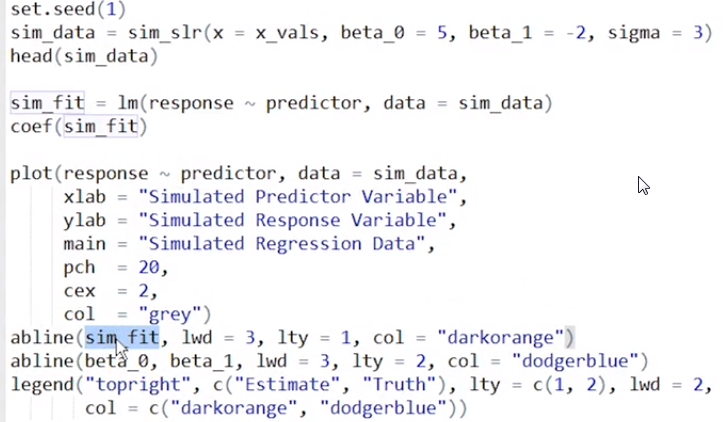
   

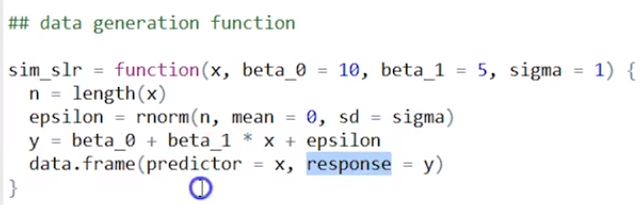
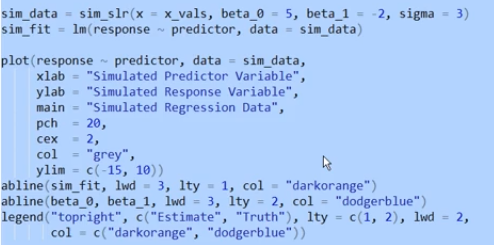
SLR Calculations in R

Simulation

In our discussion of the simple linear regression model, we said that there were three parameters: beta\_0, beta\_1, and sigma\_squared. We said that beta\_0 and beta\_1 are the signal, they control the line and sigma\_squared controls how much the observations will vary about that line - Noise parameter. To estimate beta\_1 and beta\_0, we used either the method of least squares or maximum likelihood estimation to obtain beta\_1\_hat and beta\_0\_hat. We also then said that s\_sub\_e squared is an estimate of sigma\_squared. Now, in particular for beta\_1\_hat and beta\_0\_hat, we justify these via least squares by saying these are found by literally minimizing the residual sum of squares. But now we'd like to talk about some statistical properties of these two estimates.

So now that beta\_1\_hat and beta\_0\_hat are random, we can talk about their properties as estimators. So, in this situation, when we are attempting to estimate the slope parameter and the intercept parameter of a simple linear regression model, we find that the least squares estimators, which are beta\_1\_hat for beta\_1 and beta\_0\_hat for beta\_0, are what we call the BLUW - best linear unbiased estimators. We already understand that they are estimators because we're considering them random and we're considering the process, not just applying it to a single dataset. The fact that these are linear estimators is not due to the fact that we are estimating a line, it's due to the fact that they are linear combination of the random Y\_i’s.

Unbiased - So maybe we see a bunch of beta\_one\_hats, and they sort of — they're never exactly the true beta\_1, but on average, they're essentially correct. So, while any one beta – so this is some beta\_one\_hat for a particular dataset, so it's not correct. But if we repeated this an infinite number of times and average these, they would, on average, be estimating the correct thing.

