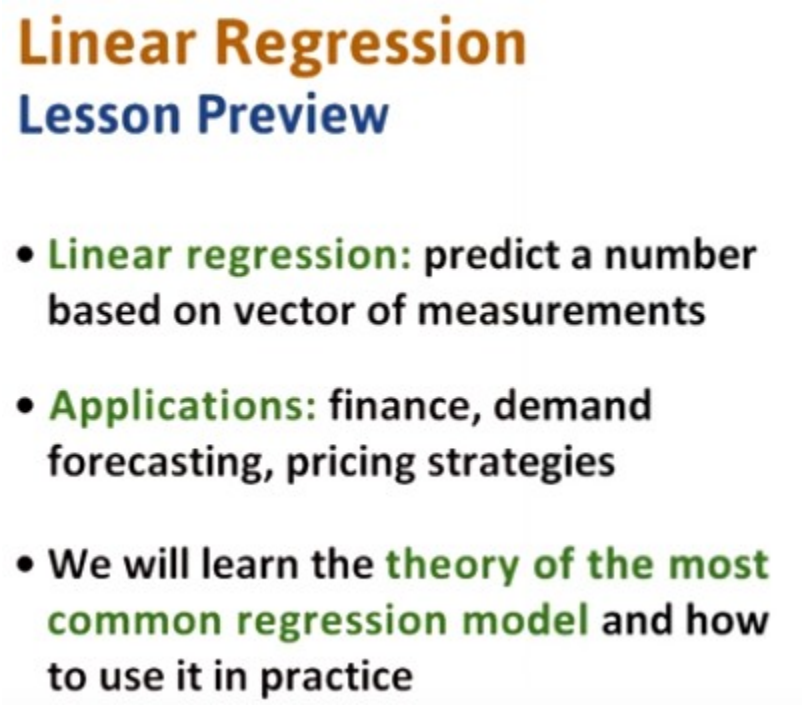
The video time for this lesson is 76 minutes.

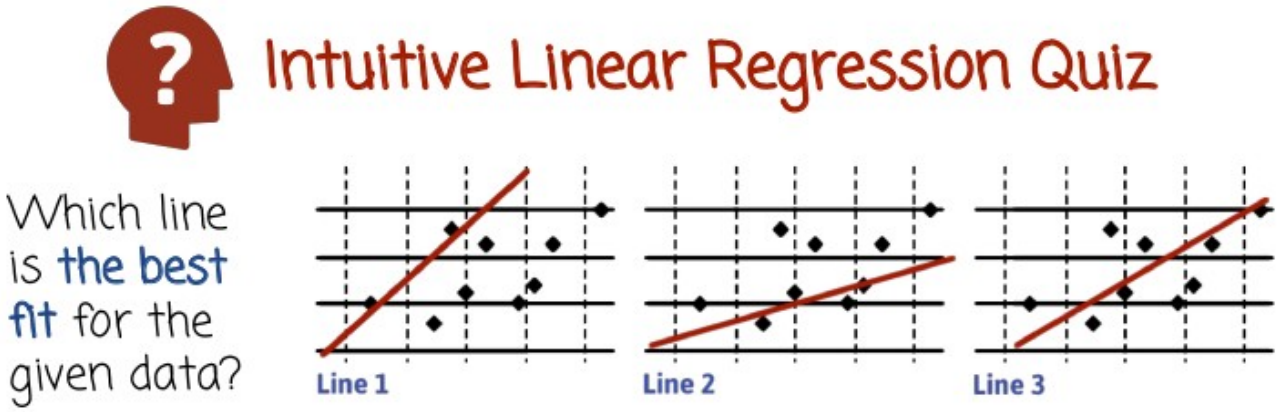
The recommended reading for this lesson is:
Linear Regression

https://s3.amazonaws.com/content.udacity-data.com/courses/gt-cse6242/recommended+reading/linReg.pdf



1. Linear regression is the task of predicting a real value or a number based on a vector of measurements. It is used widely in finance, demand forecasting, and in determining pricing strategies. It is also used in many other areas and is a key component of any data scientist's tool box. In this lesson, we will learn the theory behind linear regression and how to apply it in practice using the R programming language.

2. In this intuitive linear regression quiz, which line is the best fit for the given data? Line 1 fits the best, passes through two lines and leads to a conservative prediction. Line 2 fits the best, it touches the most points. Line 3 fits the best because it seems it is overall closer to more points.

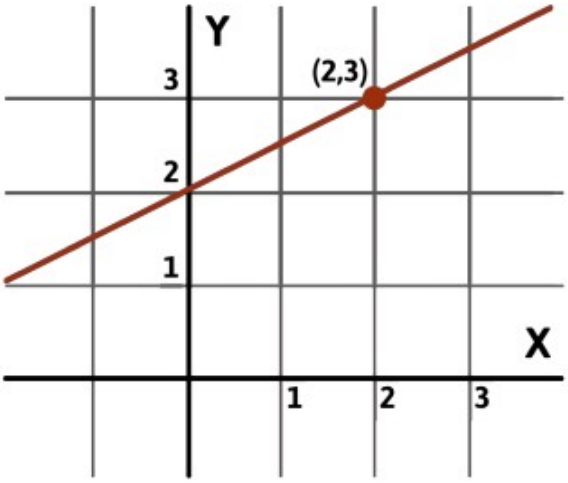


3. Well this is an intuitive quiz and we haven't yet discussed what it means to be a good fit. But we can see intuitively Line 1 looks like not a great fit because there are many points under it or to the right of it that are far away from the line. Line 2 have similar problem, perhaps less significant but above it. We see there are many points above the line, that are far away from the line. Line 3 is still not super close to all the points, but it seemed to have a good balance of points under it and over it. And as a result, it balances over prediction and under prediction. And that would be the best fit among these three lines.

4. In this quiz, filling the form here with the equation corresponding to this line. Use the slope-intercept algebraic form.

Line Quiz

What is the equation of this line?
Use the slope-intercept form.



$Y = 2 + \frac{1}{2}x$

5. The equation is Y = 2 + one-half X. 2 is called the intercept, and that corresponds to the value of Y we get when X is 0. Notice if we substitute X for 0 here, X will be 0, we get a Y value of 2, which is exactly the point where the line intersects the Y axis. Well, that's the intercept, 2. The one-half is the slope, and this corresponds to the rate of increase in Y as we increase the X unit axis. So, in this case, for every two units of increase in two units of x, there is a one unit increase in Y. Which is why the slope is one-half. If it would increase in Y by one unit for every one unit of Y, the slope would be 1. If the line would be of this form, a downward line, the slope would be negative. Could be -1 if it's a decrease of one unit in Y for an increase of in one unit in X, or it could be a different negative slope.

6. In this quiz we're going to assume that we have a linear regression model given by this line formula here. If x is 69, what might be the predicted value of Y based on this model right here?

Prediction Quiz

Given the equation of the Least Square Regression Line, predict the outcome for a given value of 'x'. $Y = -7.964 + 0.188x$

If X = 69, what might be a predicted value of 'Y'? A: 5.008

7. And the answer is just above 5 and we get that value by taking 69 and substituting it for the value x and computing this algebraic expression right here.

Meaning of Prediction Quiz

Given the equation of the Least Square Regression Line, predict the outcome for a given value of 'x'. $Y = -7.964 + 0.188x$

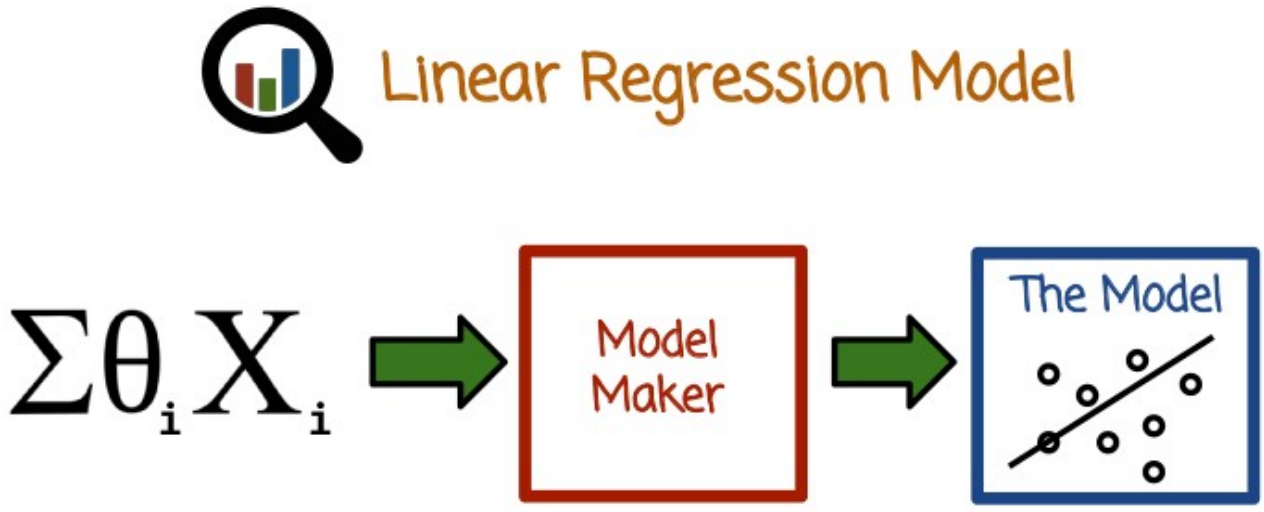
If X = 69, what might be a predicted value of 'Y'? A: 5.008

What can we say about Y? Check all that are true.

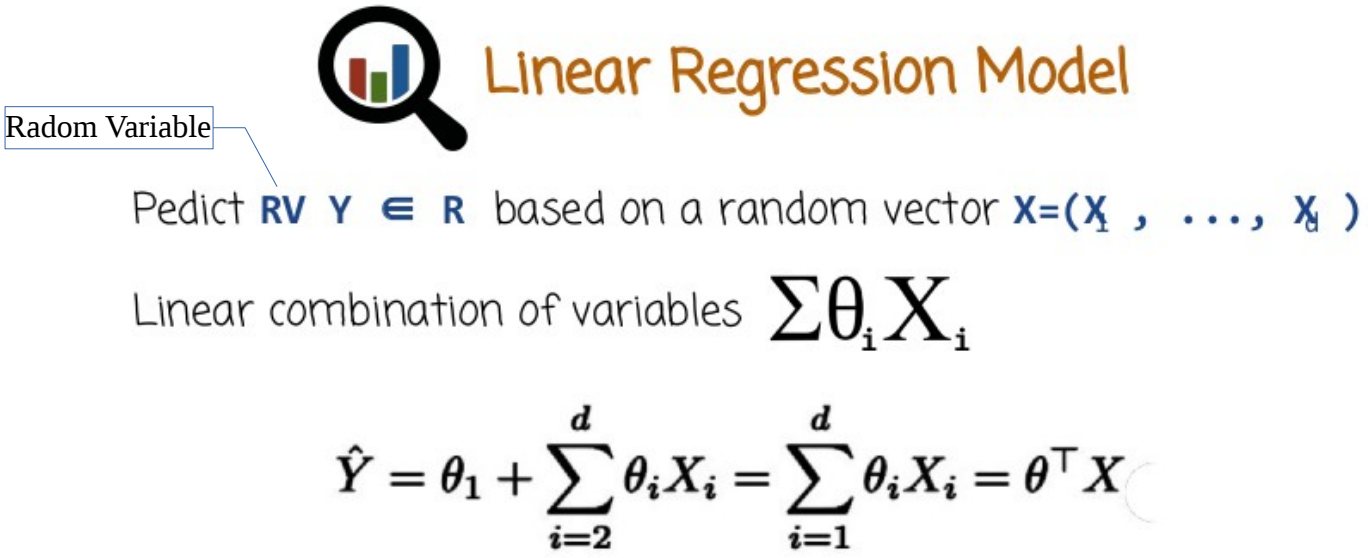
- ☐ Y should exactly equal 5.008 when X = 69
- ☑ Y could be less than 5.008
- ☑ Y could be more than 5.008
- ☐ None of the statements are correct

8. Lets see what it means to do a prediction with linear regression. Given the equation of least squares regression line, predict the outcome for a given value of x. This is the linear regression model or the line. If x is 69, what might be the predicted value of Y? Well we saw in the previous quiz here's the prediction. What can we say about Y? Check all that are true. Y should exactly equal 5.008 when x=69. Y could be less than that value. Y could be more than that value. None of the statements are correct.

9. The first statement is not correct. Because when X = 69, the prediction is that made by linear regression, is that Y should not necessarily be equal precisely to 5.008. But rather that it's a probabilistic value close to 5.008. It could be a little bit more, a little bit less, but it doesn't have to be precisely equal to 5.008. Linear regression is the statistical method or probabilistic model and it does not make specific deterministic predictions, rather it makes predictions about the probability of Y given what we know about x. So the second statement is correct, because y could be less than 5.008. The third is also correct, because it could be more than 5.008. And obviously the last one is not correct.



10. Linear regression is characterized by a formula such as this one where we have a linear combination of variables Xi and the linear combination has weights theta i and the result of the linear combination is the prediction of the linear regression model. It has a training process similar to logistic regression, where we get training data which is multiple tuples of X or vectors of X, accompanied by a sequence of labels. So for each vector X, we will have one label and we have multiple such pairs of vectors and labels. From that training data, we're going to estimate theta. That theta is referred to as the model. Once we have the theta vector, we can make prediction in the future by applying this formula to newly unseen vectors x and getting a new prediction for them.



So a linear regression model y would be a random variable, meaning it's a scalar, it's not a vector. X would be a random vector whose components are x1 through xd or a vector of random variables, and the task is to predict a random variable Y based on instantiations of the random vector X. We're going to do it based on this equation right here, theta 1 X 1 plus theta 2 X 2 and so on all the way to theta d x d. In some cases, we want to have an intercept rather than start immediately from theta 1 x 1, we want

to have theta_1 for example without any multiplication by a dimension of x. Similar to logistic regression, this gives us more flexibility in that the model can also capture situations when y is offset from zero systematically. That gives us a lot more flexibility, very useful to have this offset term here. But we still want to simplify our math and write the equations just using a sum of theta iXi and we can do that by assuming that X1 is always going to be the number 1 or the first dimension of X. All vectors X are always going to be 1. So basically augment the dataframe X with another column, which is identically 1. In which case, we can just use all the formulas that we get from assuming the model is given by this form, but in essence, we'll capture all the advantages of having this offset term right here. If we want to denote it algebraically, we can assume theta is a vector, X is a vector. Vectors we assume are normally column vectors. Theta transposed here would be a row vector, so we have here a row vector times the column vector, which is the same as an inner product of two vectors, which is precisely the algebraic form written here. Sometimes we'll see in the lesson it will be more convenient to use vector forms or vector matrix notation, so it's pretty useful to be familiar with that as well, as will the scalar notation right here.

Linear Regression Model

A probabilistic model for Y given X that assumes:

$$Y = \theta^\top X + \epsilon, \qquad \epsilon \sim N(0, \sigma^2)$$

Or equivalently

$$Y|X \sim N(\theta^\top X, \sigma^2).$$

More precisely, linear regression model assumes that the random variable Y equals theta transpose X plus epsilon, where epsilon is a Gaussian random variable with mean zero and variance sigma squared representing noise. This is an assumption that linear regression model makes, it may or may not hold in reality. Equivalently, we can state this assumption right here here, by saying that the distribution of Y condition on X is Gaussian with mean or expectation theta transpose X and with variance sigma squared.

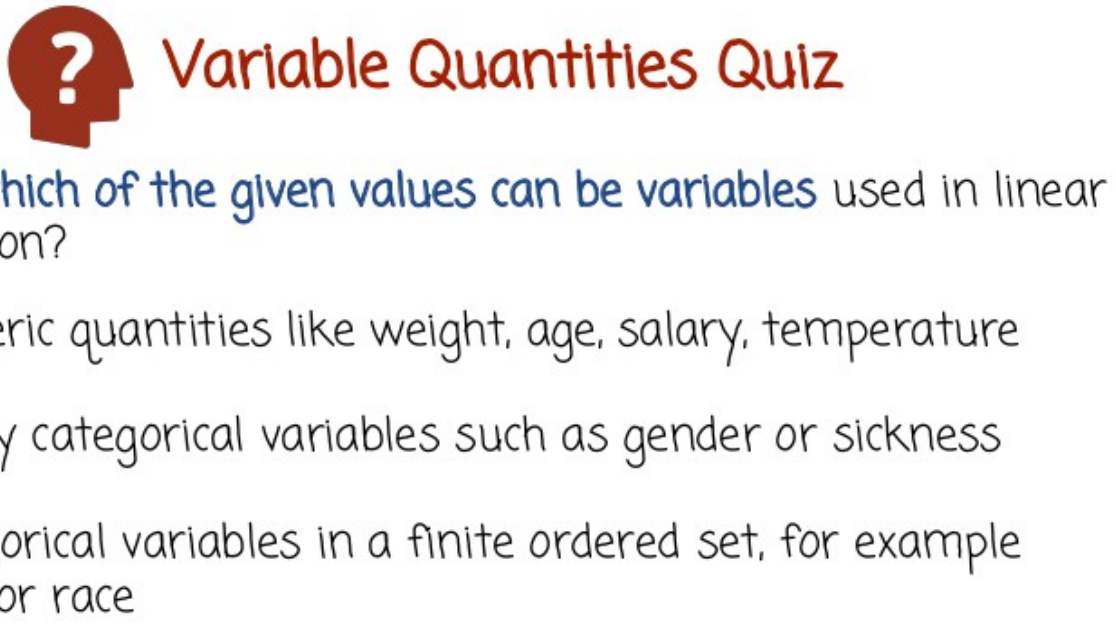
Linear Regression Model

Given **X**, **Y** is normally distributed with a mean that is linearly increasing in **X** and has constant variance.

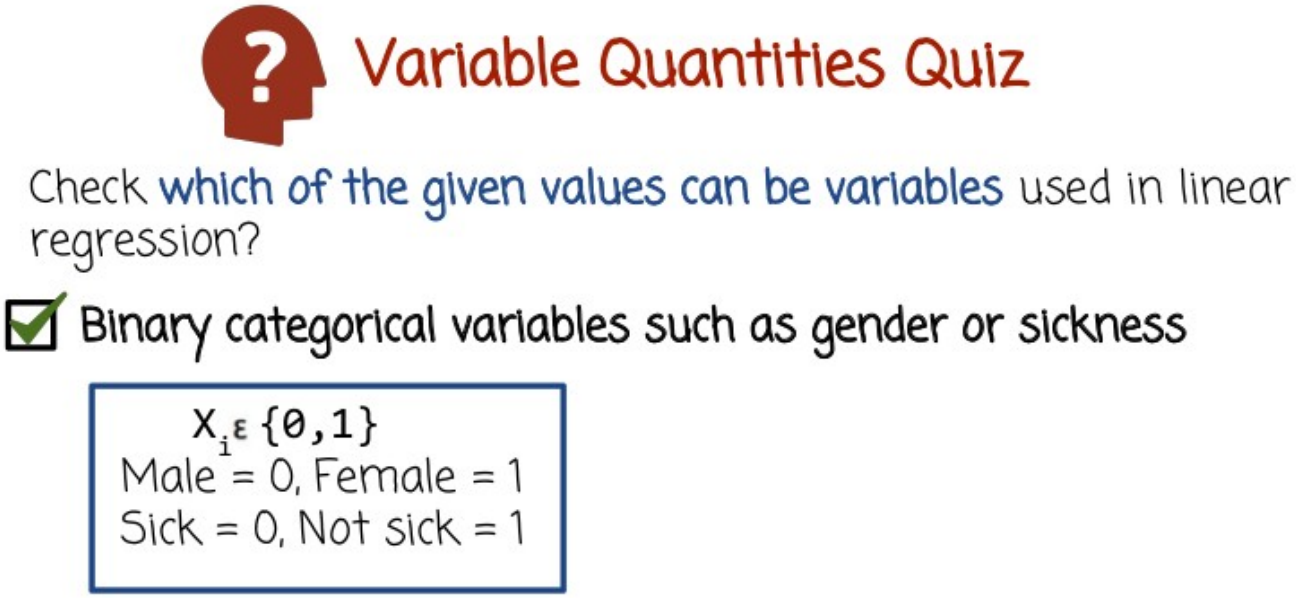
In linear (and other) regression model, no assumption is made on the distribution **p(X)** and we do not attempt to model **p(X)**. Rather all effort is focused at **p(Y|X)**.

In other words, given X, Y is normally distributed with a mean that is linearly increasing in x if x is a scalar, if x is a vector it would depend on the inner product theta transpose x and has constant variance. It's important to understand that in linear regression and other regression models, typically no assumption is made on the distribution p(x) and we do not attempt to model it. Rather all the effort is focused on the conditional distribution p(y|x). So linear regression again makes an assumption on the conditional distribution. It is agnostic or does not make any assumption about the distribution p(x) and which can be arbitrary. And we're specifically not trying to model p(x). As a result, when we train or obtain the linear regression model, we can effectively predict the value of Y from vectors X, but we're not able to predict the distribution of X.

11. Check which of the given values can be variables used in linear regression? Numeric quantities like weight, age, salary, temperature. Binary categorical variables such as gender or sickness. Categorical variables in a finite ordered set, for example, color or race.

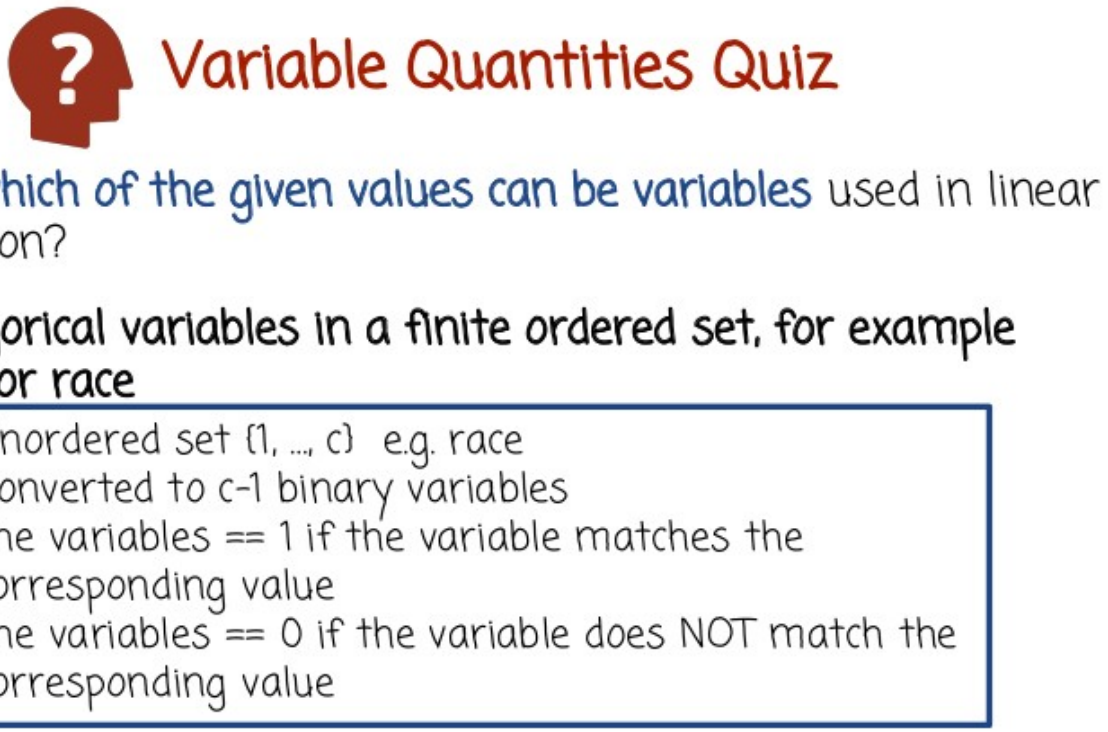


12. The first statement is correct, and we can just use numeric quantities by just plugging in their values in the linear regression formula that we saw before. Or we can transform them by using different transformations like log or some power or square power and so on on these numeric quantities. The second statement is also correct. We can use binary categorical variables such as gender or sequence. The third statement is also correct. We can use categorical variables in a finite ordered set, for example, color or race, or many other examples exist.



Now let's see how we do that. In the case of binary categorical variables such as gender or sickness, an easy way to represent it is to just map the value of the binary variable yes to 1, 0 to no, or the other way

around. So let's say if we want to capture gender, we can map for example male to 0, female to 1, or the other way around. And this value will be the value of the feature or the random variable xi that will be plugged into the linear regression formula and multiplied by the corresponding theta.



In the case of categorical variables in the finite ordered set, for example, color. We have a set of values 1 through c, could be white, black, brown, green, blue. Now there are several ways to convert that into features or variables in linear regression. One way is to convert that categorical variable into c-1 binary variables. The variables will be 1 if the value of the original categorical variable matches the specific index of that variable within that vector of binary variables. And if none of the c-1 binary variables are 1, they're all 0, then the last value holds. 下面是例子.

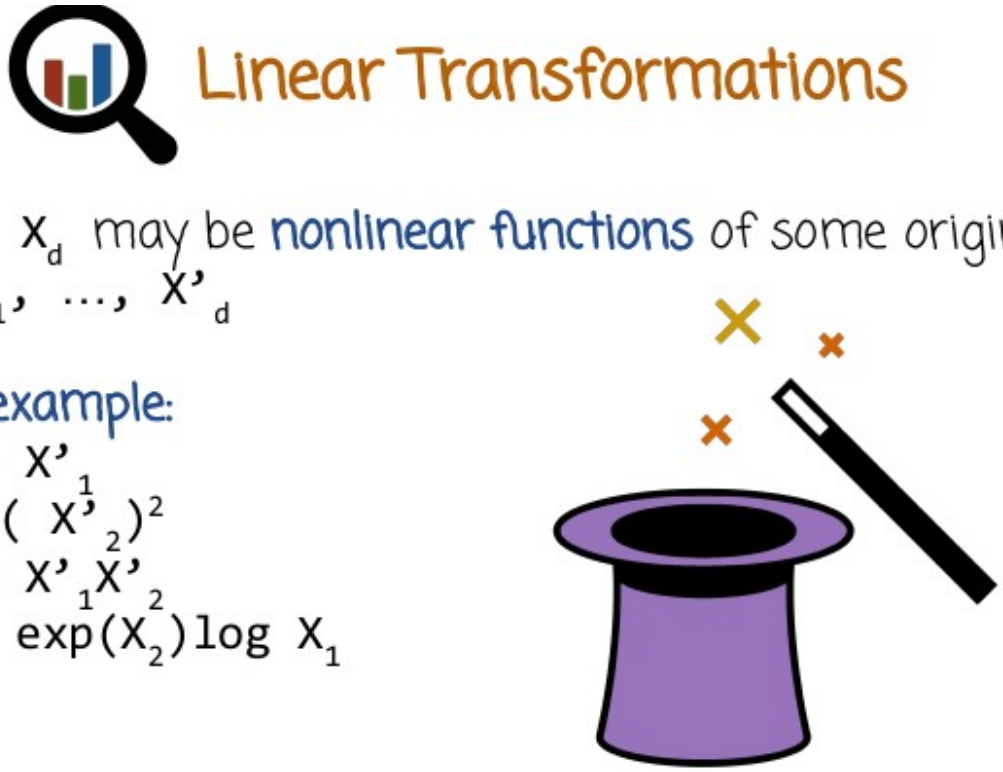
Variable Quantities Quiz

For example:

Dog Breeds	Corresponding value = Poodle	Corresponding value = Bulldog
Bulldog	0	1
Beagle	0	0
Poodle	1	0

上圖中的第二列就是一個 X^(i) vector, 即(0, 0, 1). 第三列也是一個 X^(i) vector, 即(1, 0, 0).

So let's see an example. Let's suppose we have a categorical variable which has four possible values, bulldog, beagle, poodle, and other. So if the result of the variable is poodle for a specific instance, then in your regression we can represent that with a vector of 3 binary values, 0, 0, 1, where poodle corresponds to the third variable, beagle the second and bulldog is the first one. If we have another instance where the variable of the categorical variable is bulldog, we can represent that with a three dimensional binary vector here, 1, 0, 0 and 1 at the first spot is going to represent the value that bulldog is correct, 0 and 0 meaning that beagle and poodle are not correct. If we have an other, or neither bulldog, beagle, or poodle, then we can have 0, 0, 0 representing the other value. Now we take this vector of three binary values, and we just insert it along with the other variables in the linear regression formula. (本段剩下的是廢話且容易不知所云, 不要看) So, for example, if we have one categorical variable with, let's say, a thousand different values, that could be words in a document, for example. If there are a thousand different values, there will be a thousand different binary variables, and they will be added together with the other features and variables in the linear regression model. If we have 100 categorical variables, each one of them having 100 different values, we are going to get a binary vector of size 10,000, or dimensionality 10,000 and that will be added to the linear regression model together with any other variables.



13. So linear regression has one major weakness and that is that the relationship between Y, and X is constrained to be linear. As we saw with logistic regression, that's not necessarily as big limitation as it seems. Because what we can do is we can take the original data, which here we call X prime and transform these features or variables using nonlinear transformations into a new vector X1 through Xd. Now the new vector, which is non-linearly related to the original vector may be of the same dimensionality as the original vector or may be higher. Let's see a few examples. X1 may be X1 prime. X2 may be X2 prime to the 2nd power squared. X3 may be X1 prime times X2 prime and X4 may be e to the power of X2 squared times log effect X1. So in this case, the original vector is of dimensionality 2. X1 prime X2 prime, then new vector is of dimensionality 4 and is non-linearly related to the original vector.

Linear Transformations

$x_1, \ldots, x_d$ may be nonlinear functions of some original data $x'_1, \ldots, x'_d$

The linear regression model $\theta^T X$ is linear in x but nonlinear in the original data x'.

But the linear regression model, theta transpose X is linear in X, the transformed values, but is nonlinear in the original data X prime.

Linear Transformations

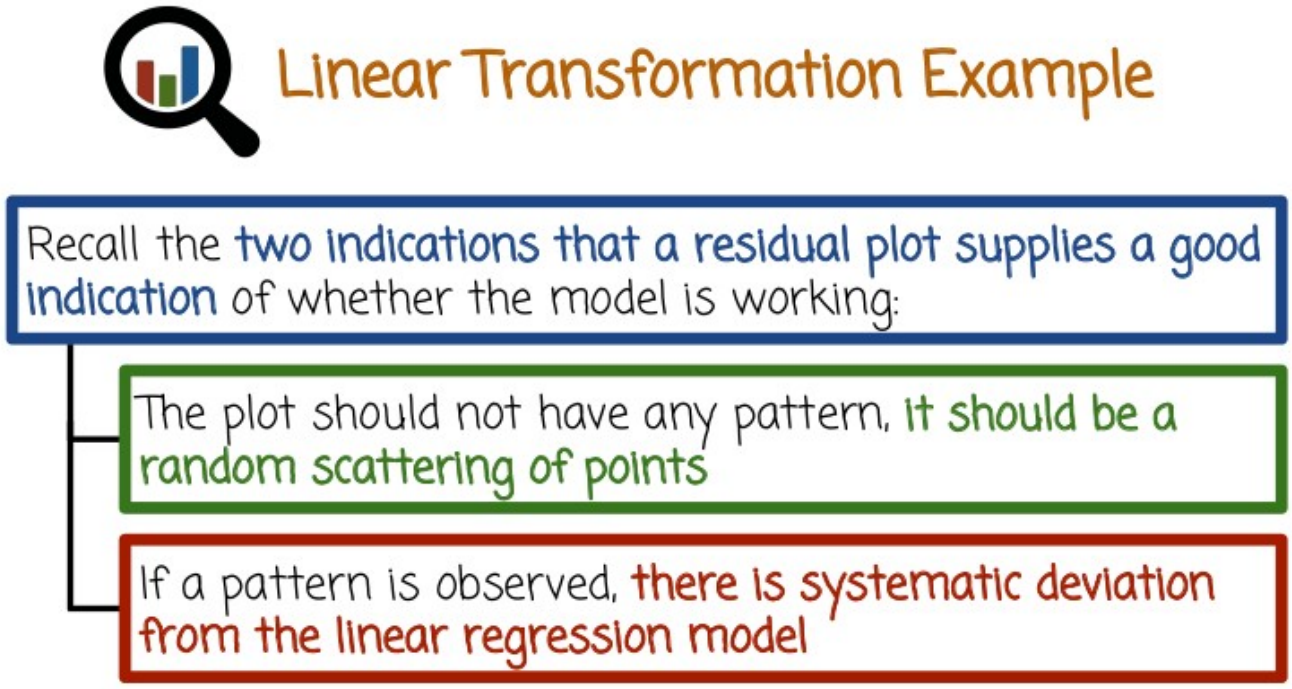
$x_1, \ldots, x_d$ may be nonlinear functions of some original data $x'_1, \ldots, x'_d$

The linear regression model $\theta^T X$ is linear in x but nonlinear in the original data x'.

The result: Substantial flexibility to model non-linearly related X and Y

The requirement: A nonlinear transformation $X' \rightarrow X$ for which a linear relationship exists between Y and X.

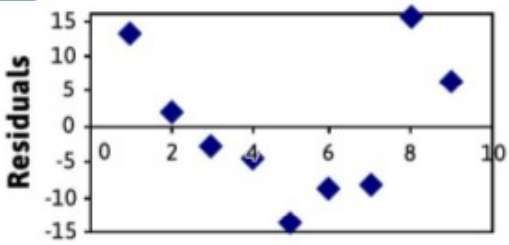
As a result, we get substantial flexibility to model non-linearly related X and Y even though we used linear regression. To do so, we need a nonlinear transformation from X prime to X for which a linear relationship exists between Y and X or for which a linear relationship approximately exists. And if that linear relationship is more present between X and Y, than between X prime and Y, then that's certainly something worth thinking about.



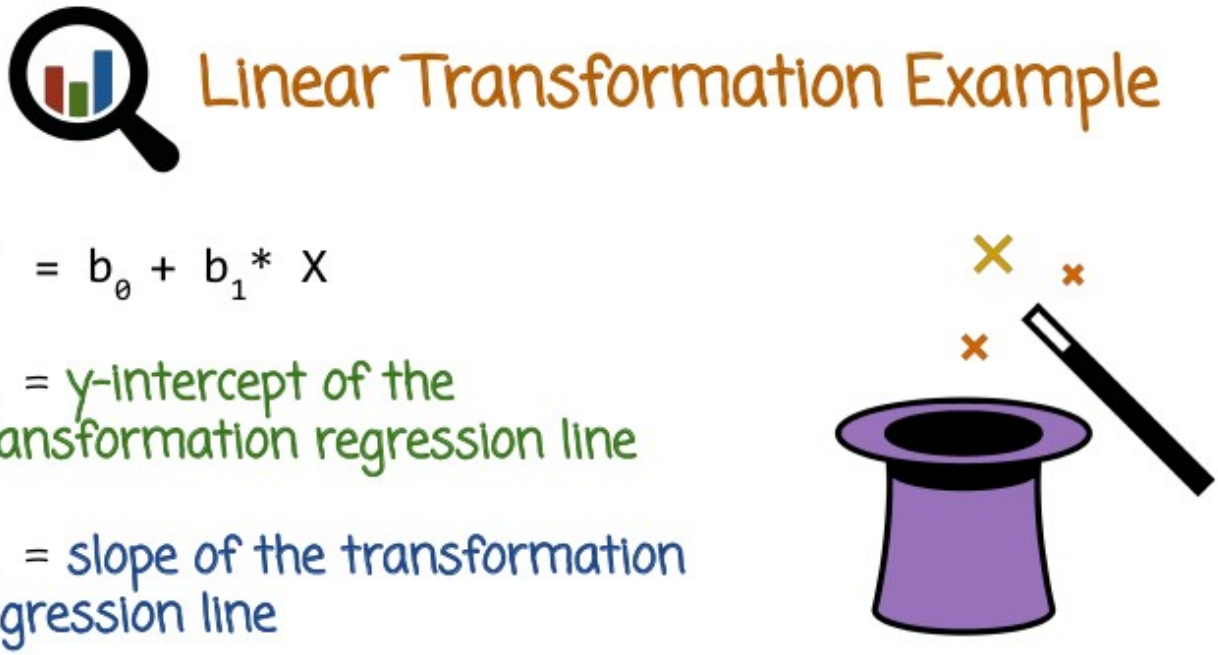
14. We're going to see a simple example in a second. But before that, I want to talk about something called a residual plot. A residual plot is a plot where x marks the coordinate of the example and y marks the residual, which is the relationship between the original value and the prediction of the model. 對 residual 的意思更好的講解 → Usually, what we do is just take the difference between the original value and the predicted value. That residual plot provides a pretty good indication of how well the model is working. If the residual plot has very significant or clear pattern, then that probably shows that there's a systematic signal that is not being modeled by the linear regression model. If the residual plot seems random, it may mean that the linear regression model works pretty well or as well as it can and the resulting residuals, which may be even big, meaning the model is not a very accurate model, are the result of some inherent noise. Of course, it's possible that the residual plot is not as easily understood and there is a systematic pattern to the points that's just difficult to predict or to notice using just a visual inspection. But in many cases, a systematic deviation would be actually quite visible and it's certainly worth looking at the residual plots.

Linear Transformation Example

x	1	2	3	4	5	6	7	8	9
y	2	1	6	14	15	30	40	74	75

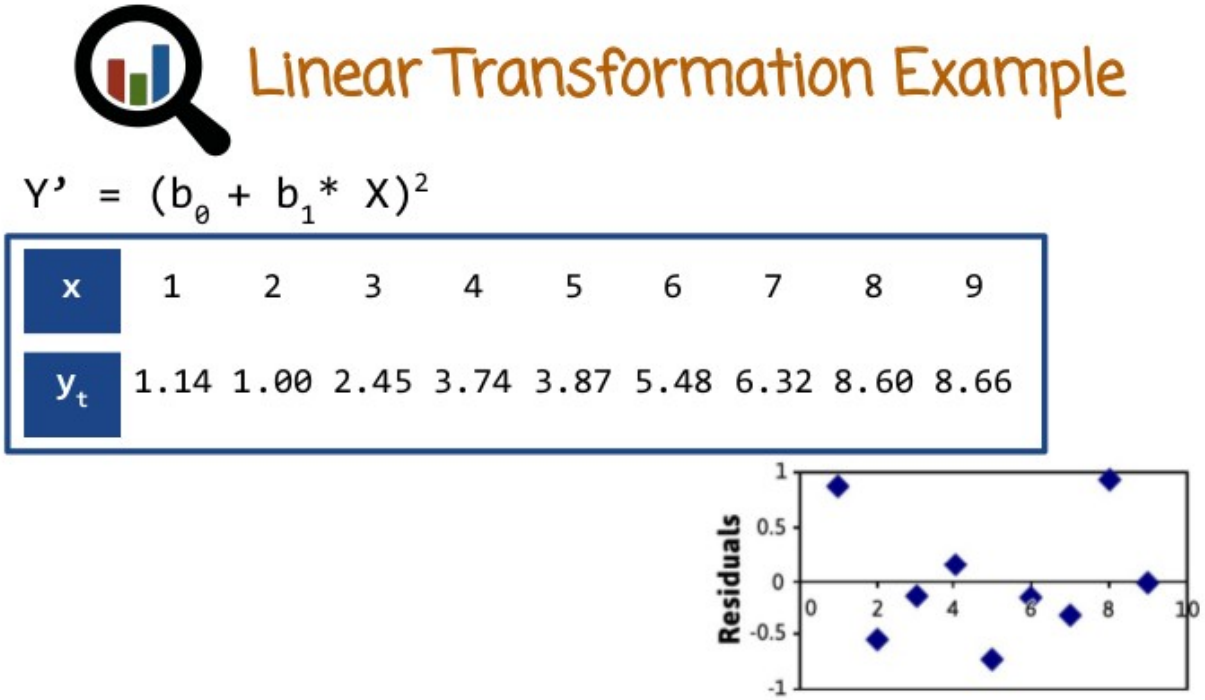


So, let's see an example. Here are values of x and here are values of y, and we can assume that there is a specific regression model being fit to these data, and the resulting residuals are shown by this graph right here. So we have nine different points, because we have nine different axis and the y-axis of each point is the difference between the original value and the predicted value of the model. And we see here, a systematic pattern of values going down and then going up, which is something that the linear regression model did not capture. And it may make sense to go back to the linear regression model and modify it and try to get a better fit. So in this case, what we're going to try and do is non-linearly transform the data and then fit again a linear regression model and look at the resulting residuals.



上圖中的 Y' 是原來的 model.

So, let's consider a transformation of the variables and fit a new linear regression model.

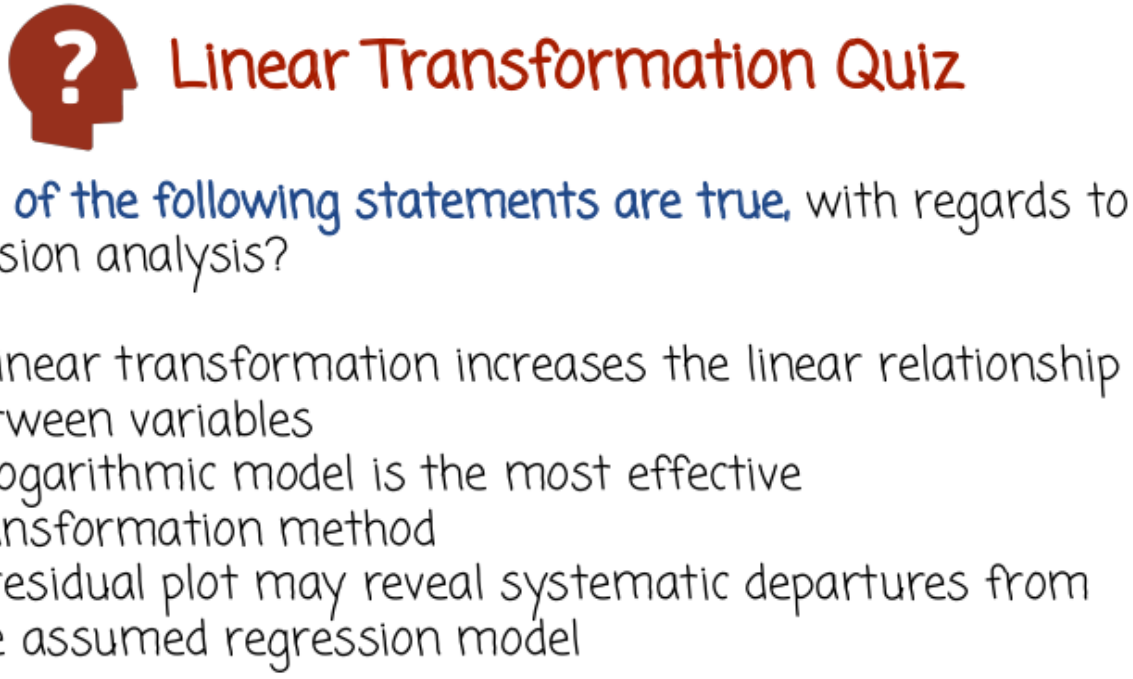


上圖中的 Y'就是新的 model, 展開後為 Y' = b0^2 + (b1 X)^2 + 2 * b0 * b1 * X. 其暗含的 non-linear

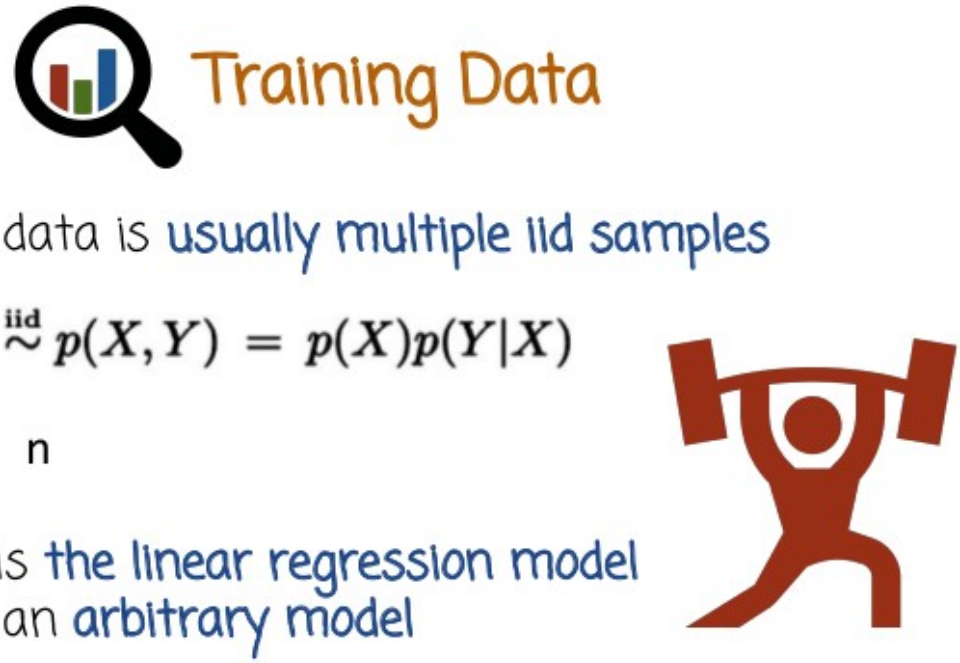
transformation 為: {x} → {x^2, x}.

In this case, what we do is we have (b0+b1 X)^2. If you expand this square, you get b0^2 + (b1 X)^2 + 2 * b0 * b1 * X, hich corresponds to just adding a 2nd power to the linear regression model. So you have the intercept, you have a linear term and then you have a square term. So basically, we add in another dimension. And the residuals that we get are now, they look like they're much more random or there's less of a systematic pattern in the residuals. So, it looks like the transformation here helped us get a better fit.

15. Which of the statements are true, with regards to repression? The linear transformation increases the linear relationship between variables. A logarithmic model is the most effective transformation method. A residual plot may reveal systematic departures from the assumed regression model.



16. So, the first statement is not correct. Linear transformation increases the linear relationship between variables although in-non linear transformation may. The second statement is also not correct it may or may not be that logarithmic model or transformation is a good transformation in many cases useful but that's not necessarily so. The third statement is correct, the residual plot may reveal systematic departures from the assumed regression model.



17. So, the training data for linear regression is a collection of pairs of vectors accompanied by labels y. Y would be the response variable, it's a scalar, a random variable and X would be a vector of features or measurements. This superscript i here refers to the fact that we have several of them. i goes from 1 to n. And so we have n vectors and n labels. Usually we arrange them in a data frame where every row would be the first columns of that row would correspond to the vector x. And the first or the last column would correspond to the value y. And i would correspond to the row numbers. So as you go down the different rows, you'd get different pairs of vector of measurements accompanied by labels. Now, these pairs are sampled from a joint distribution over x and y, and we can rewrite that joint distribution as p of x times p of y, given x. Remember p(x) is the model that we do not assume anything about. We do not model it. We do not assume it to be a specific form. But the linear regression model does assume that p of y given x is given by the linear regression model, which means normal distribution with an expectation or mean of theta transpose X and a variance of sigma squared. Now this is an assumption. The actual training data may be sampled from a distribution here. And p(Y|X) is not the Gaussian distribution I just mentioned. We nevertheless can still make the assumption that it is, and move forward with the linear regression model. If the assumption is correct, meaning if the data is generated from a distribution where p of Y given X is in the linear regression family or is a linear regression model for some theta vector, then we have theoretical guarantees that the training process will produce a vector of theta that gets closer and closer to the vector of theta that was use to generate the data as the dataset size or n grows to infinity. I want to emphasize again that this is an assumption, and we can still use linear regression. And in fact, in most cases, the assumption is not correct, but we can still use linear regression model, which does make this assumption, even if the data is not generated from that model. The deviation between the assumption and the distribution that a data is actually generated from may be responsible for having the model perform poorly or well in practice.

Training Data

The training data is usually multiple iid samples

$$(X^{(i)}, Y^{(i)}) \overset{\text{iid}}{\sim} p(X, Y) = p(X)p(Y|X)$$

	p(X) refers to:
Observational Data	nature
Experimental Data	Experimental design

There are usually two ways of getting training data. The first way is where we just observe the pairs of x and y without any intervention on our side, and this is called Observational Data. In this case, X is sample from p(X) and p(X) refers to some, we call it sometimes nature or some process that generates the data which we're not in control of. The other case is called Experimental Data where we are able to interfere or intervene or set the values of x as we wish. In which case x is samples from p of x, and we control what p of x is, or someone else controls. That process of controlling what is p of x is called experimental design. We can do it in some cases. If for example, we want to model the output of crops, in different regions, we can decide where to plant crops and then we can measure them later down the road. And the process of deciding where to plant the crops in the field, which locations, would correspond to the construction of p of (X) and then Y would be the outcome of the crops or the production in these locations. So if we are able to plant the crops and measure them later, then they'll be experimental design or experimental data. Whereas if we just arrive at the place and we weren't involved with where the crops were planted, rather, someone else planted them, or some other way we arrived at the particular point and we are just able to measure, then the data is called Observational because we do not have the power to modify or to specifically evaluate y at specific points.

Training Data

Training Data	Training Data in Matrix Form
$(X^{(i)}, Y^{(i)})$ $i = 1, \ldots, n$	$\mathbf{Y} = \mathbf{X}\theta + \varepsilon$ $\mathbf{Y} = (Y^{(1)}, \ldots, Y^{(n)}) \in \mathbb{R}^{n \times 1}$ $\mathbf{X} \in \mathbb{R}^{n \times d}$ $\mathbf{X}$ is a matrix whose rows are $X^{(i)}$ $\varepsilon \sim N(0, \sigma^2 I)$

ε is the vector of noise values
$\varepsilon_i = Y^{(i)} - \theta^T X^{(i)} \sim N(0, \sigma^2)$, $i = 1, \ldots, n$ corresponding to the training data and is therefore a multivariate normal vector

18. In matrix form we can express the linear regression assumption or model in the following way. Y = X theta + epsilon, where Y is the concatenation of all the labels that's within the training data. Y is superscript 1 to Y superscript n. There will be a column vector. X will be a matrix in this case, where the rows are different instances and the columns are different dimensions or variables. And epsilon is a Gaussian vector, or it's a vector with a Gaussian multivariate distribution with expectation vector 0 and covariants matrix sigma squared times I, I being the identity matrix. So this is the matrix or vector notation of the data that we observe assuming the linear regression model. And if you try to parse it, by remembering that a matrix times a column vector is given what it produces, it's just inner product of the rows of X with the vector theta. Then that gives us exactly the linear regression form we saw before in scalar form applied to each one of the components of Y, or epsilon, or the vector resulting from x times theta. This matrix notation is used often in statistics books and in papers, it's useful to know. And it's also useful because sometimes we need to do, derivations and the derivations are actually simpler in matrix form rather than in scalar form.

Training Data

Training Data in Matrix Form

$$\mathbf{Y} = \mathbf{X}\theta + \varepsilon$$

$$\mathbf{Y} = (Y^{(1)}, \ldots, Y^{(n)}) \in \mathbb{R}^{n \times 1}$$

$$\mathbf{X} \in \mathbb{R}^{n \times d}$$

$\mathbf{X}$ is a matrix whose rows are $X^{(i)}$

$$\varepsilon \sim N(0, \sigma^2 I)$$

The matrix **X** and vectors **Y** and ε correspond to the n training set instances

So a more compact way to represent the linear regression model applied to the data, which is a matrix or sequence of row vectors x arranged in a matrix and a column vector Y, is to say that the vector Y condition on the matrix X has a multivariate Gaussian distribution with expectation vector X times theta. X being a matrix, theta being a column vector and a covariance matrix, sigma square which is a scalar times the identity matrix.

Minimizing the Sum of Square Deviations

RSS = residual sum squares

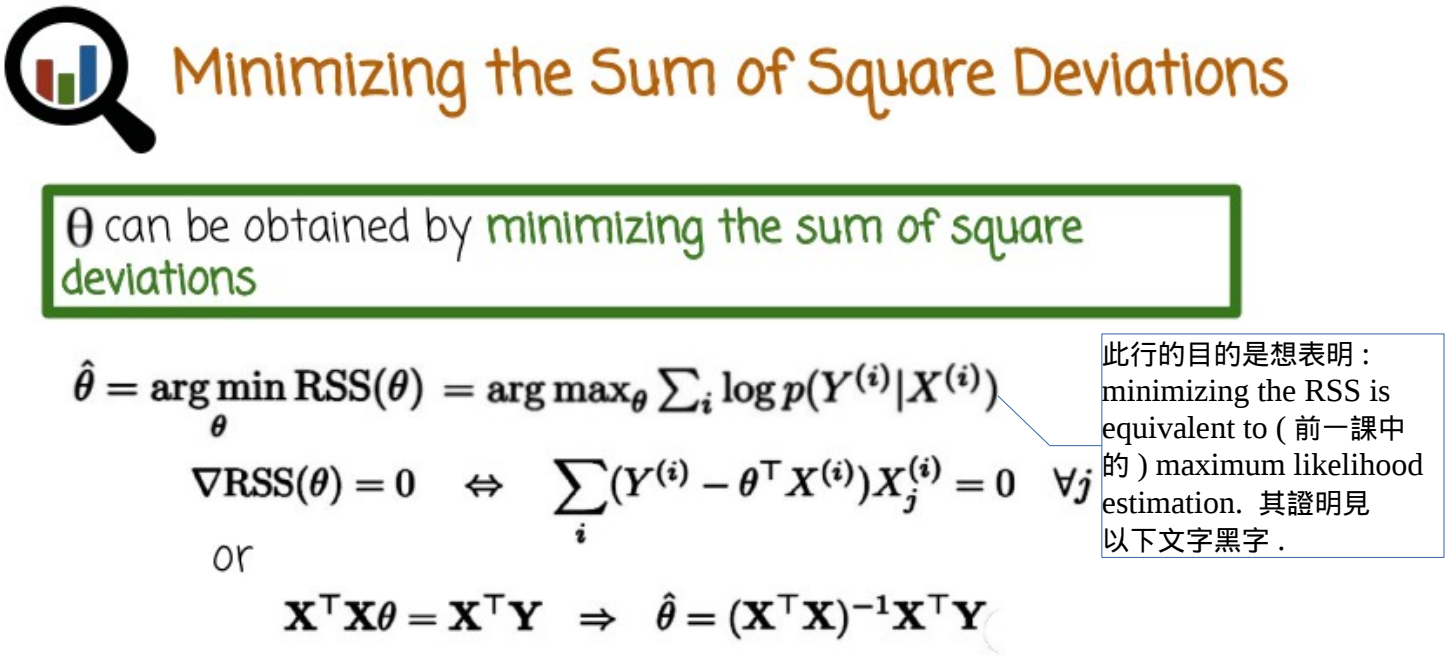
θ can be obtained by minimizing the sum of square deviations

$$\hat{\theta} = \arg\min_{\theta} \mathrm{RSS}(\theta) \text{ where}$$

$$\mathrm{RSS}(\theta) = \|\mathbf{Y} - \mathbf{X}\theta\|^2 = \sum_{i=1}^{n} (Y^{(i)} - \theta^{\top} X^{(i)})^2$$

19. So let's see how we get the vector of theta. So in linear regression we have this concept called residual sum of squares, briefly expressed as RSS and the traditional way of getting theta is by minimizing that expression minimizing the RSS as the function of theta. So the RSS, will see it in a

second what it is, but it's the function of theta. And what we're doing is we're finding the theta or the theta, theta is a vector by the way so remember that. So we're finding the theta vector that minimizes or achieves RSS with the lowest value and that theta, we're going to mark it with theta hat. And that's going to be the result of the training process, and the vector that can be used for prediction later on. So RSS expressed in matrix notation is the L2 norm of a vector which is the vector y- x times theta, that's the notation we saw before. Y is the vector of the labels arranged in the column vector. And x is a matrix whose rows are the measurement vectors and theta is the model vector that we're trying to obtain. The result of this is a vector, this is vector, the subtraction of two vectors is a vector the L2 norm is just the sum of squares of components of that vector so we want to write it. In a scalar format this is what we have here, a sum i goes from 1 to n over all training examples, we take the square between the ground true for the variable data, the variable label. For that particular instance and the value that the model predicts. That measures some goodness of fit, so the sum of the squared residuals, if you remember residuals is the distance between the value and the true value. We take the square of that because when we sum it we don't mistakes on two different sides positive and negative to cancel out. We want to measure mistakes whether they are positive side or negative side and have them accumulate, rather than have them cancel each other. So we square them, which also has the interpretation of penalizing larger mistakes more than smaller mistakes, and then we sum up all these mistakes, and that is the criteria that we're going to try and minimize. When we minimize that we will get a linear regression model given by theta that is relatively good fit and will give us relatively good prediction of future y's for new axis.



此行的目的是想表明：minimizing the RSS is equivalent to (前一課中的) maximum likelihood estimation. 其證明見以下文字黑字 .

20. So we saw earlier in logistic regression the method of maximum likelihood. And we see here that the training method that I just described minimizing the RSS is equivalent to maximum likelihood estimation. The reason being is that if we write down the maximum likelihood in this case, maximum log likelihood. Remember logistic regression maximizing the log likelihood is the same as maximizing the likelihood. And the log of the products is the sum of the logs. Therefore, the log likelihood can be written this way. And now remember linear regression model assumption is that Y given X is Gaussian with a mean of theta transpose X. And if you substitute here in that expression right here, the density of a Gaussian distribution which involves an exponent and so you have the log of the exponent, they

cancel out each other. And you just have the square deviations which is exactly the RSS. And down comes a negative or a minus sign, which converts the arg min to an arg max. In other words, if you maximize the likelihood or the log likelihood for a model corresponding to the linear regression model, Y given X being a Gaussian distribution, you simplified the expression right here. Taking the log of the density you get exactly minus of the RSS which corresponds to minimizing the RSS. And the reason I'm pointing at this out is to tie the training method to maximum likelihood because in logistics regression lesson we saw several theoretical strategies or nice properties of maximum likelihood. Specifically consistently that the theta that we get will converge to the real theta as n goes to infinity. Assuming that the data is really generated from a linear regression model. And efficiency that that convergence is at the best possible rate. Okay, but let's see how we do that in practice. How do we either maximize the log likelihood or minimize the RSS, which are equivalent? How do we actually do that? And so one condition for arriving at the stationary point, minimum or maximum is that the gradient is 0. So the gradient of the RSS with respect to theta, same thing with the log likelihood. Now in the case of logistic regression, we had to come up with an iterative process called gradient descent, which iteratively estimates the gradient and follows it In order to get to the maximum or minimum. In the case of linear regression, the least squares form of the RSS is simpler than the likelihood of the logistic regression. And as a result, we can try to solve this equation explicitly and see when does the gradient equals 0, and actually solve it in closed form. And that will help us get to more efficient algorithms for training logistic regression. Now if you write down the residual sum of squares, and compute the gradient, the partial derivatives with respect to different components of theta and set each one of them to 0 and you write it in matrix form. You can get exactly this set of equations right here. This is a set of the equations. One for every component of theta. And if we want to write it in matrix format, we can write it this way. So the set of equations describing the condition that the gradient is 0 is written here as the different equations. But if we want to write them as a single vector equation we can write it in this way. And we can verify that this is the same as this by expanding it and doing some algebra. Now remember I said there's an advantage to writing it in matrix or vector form. The advantage is that you can actually isolate theta. That's what we're trying to do. We're trying to solve this equation for theta. And X transpose X is a matrix, and that multiplies theta from the left. So we can just multiply both sides of the equation with the inverse of X transpose X. So X transpose X inverse multiplied by X transpose X is the identity matrix, or just theta. And then we also have to do it on the right-hand side of the equation, X transpose X inverse multiplying X transpose Y. And what we get is a solution to the equation, and therefore we label this theta hat which is our maximum likelihood estimator or residual sum of square minimizer. And so what we have is a very explicit or closed form formula that relates the data. Remember, X is a matrix whose rows are the data vector and Y is a vector whose rhos are the labels. So this formula relates the data that we observe, the training data, precisely to the vector theta hat. Looking at it this way it looks like there's no need to do an iterative process like gradient descent. We just compute X transpose X, you invert it, multiplied by X transpose, multiplied by Y. You get your vector theta hat and that is the linear regression model. That is true in principle, but in practice the matrix X transpose X would be relatively large. And inverting it could take a long time, and so we may still need some iterative process to solve it. Nevertheless it could be faster than gradient descent. We're not going to get into how this iterative process works, but it's important to understand if the dimensions are not high the matrix X is a relatively small matrix. We can solve it explicitly without any iterative processes just by computing this expression right here but in cases of larger data that's going to be a problem and we still need an iterative process. But the iterative process can be the iterative process required to insert in the matrix rather than the iterative process required to do gradient descent which is simpler. So we can basically conclude that training linear regression is simpler and faster than training logistic regression.

Minimizing the Sum of Square Deviations

$$\hat{\mathbf{Y}} = \mathbf{X}\hat{\theta} = \mathbf{X}(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\mathbf{Y} = \mathbf{H}\mathbf{Y} \quad \text{where} \quad \mathbf{H} = \mathbf{X}(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}$$

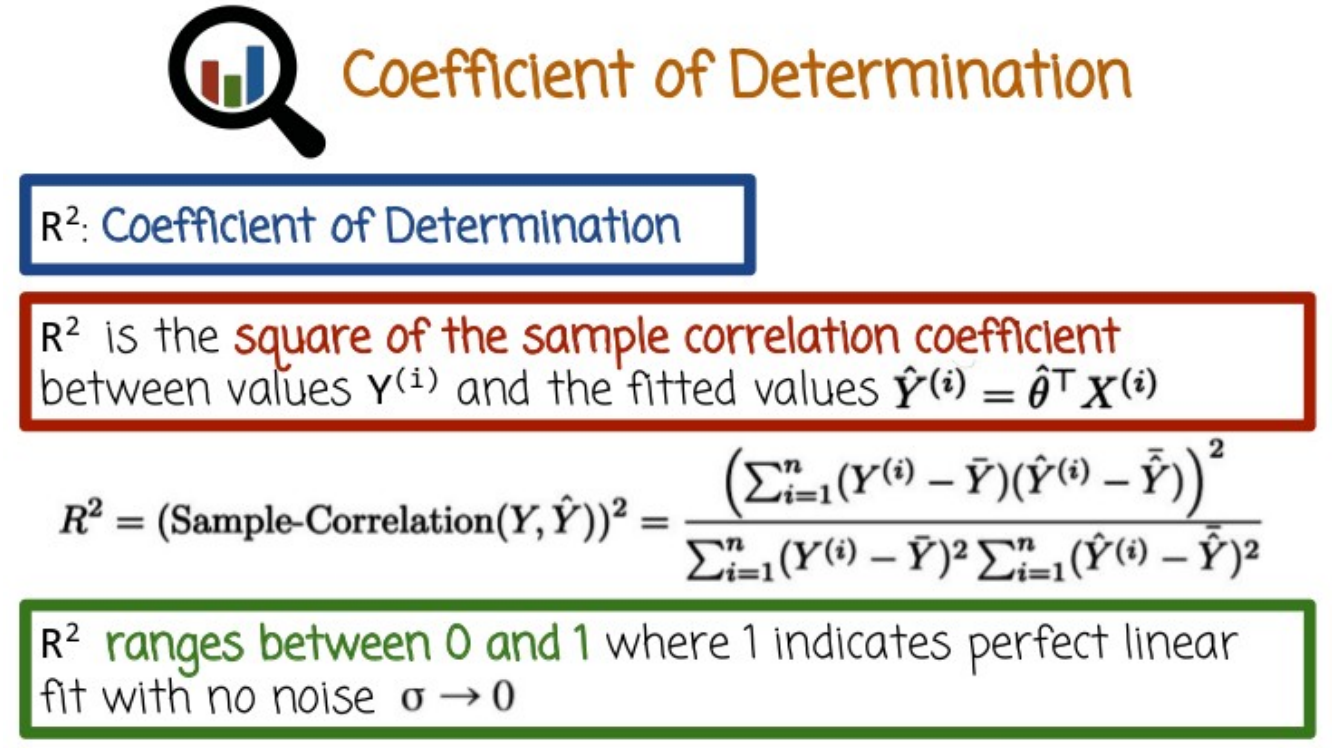
Special Case: the columns of **X** are orthogonal

$$\hat{\theta}_j = \langle u_j, \mathbf{Y} \rangle / \|u_j\|^2$$

上圖框中的不重要, 看看就是, 此時 X^T X = I, θ^j 中的 j 即第 j 個 instance, u_j 即為 x 的第 j 列.

21. We can do some more algebra. Looking at the equations that we had in the previous slide, we want to relate now the predicted values to the data, which is basically the matrix X and the vector Y. The vector Y hat is the predicted values by the model associated with the training observations. And so we can substitute, instead of theta hat, we substitute the formula for theta hat that we saw before and we get this expression right here. And we can just note that X times X transposed X inverse times X transposed. This is a matrix, which we can just call H. We called also the hat matrix. And when we take that H or the hat matrix, which is composed entirely of the matrix. We take that matrix H and we multiply by the labels Y, that gives us the prediction Y hat. So that's a very nice and simple relationship between the predictions, and the actual observations. And this can be used to do a lot more analysis on linear regression and derive more properties, and do more theoretical work in order to understand the properties of linear regression. One special case is when the columns of X are orthogonal (正交的). Now this doesn't normally happen, but it's kind of useful to see what happens when it does happen, because it simplifies the expression and it gives us insight and also enables us to do theory and understand things you wouldn't normally see as easily. So if it so happens that the columns of X are orthogonal, then X transpose X is the identity matrix and things simplify to the point where if you look at the previous formula in the previous slide for how to compute theta hat. Theta hat, the js component is simply the inner product of uj and Y divided by the normal of uj. In this case, there are vectors u are the columns of X. Remember, they are orthogonal. And what this formula is showing us is that we can see in this special case, the u js are the columns of the matrix X and we can obtain theta j hat by taking Y, the vector Y and projecting it onto the columns of the matrix X doing orthogonal projections.

22. So we saw previously one way to diagnose the model or evaluate the fit of the model, which is by looking at the residuals. Another way is to look at the R squared, also known as the coefficient of determination. R squared together with the Residual Sum of Squares measure the model quality. Residual Sum of Squares being just the sum of the squared residuals.



28 段上面會介紹 correlation coefficient, 跟這裡的的 R^2 不同. R^2 不能為負, 而 correlation coefficient 可以為負(如-1).

Specifically, R squared is the sample correlation coefficient between the original label values Yi given

in the training set and the fitted values Yi^ given by the prediction of the model on the training data. The formula for the correlation coefficient, in this case, the R squared expression is given here. This is simply the fraction, where the numerator is the co-variance and the denominator is the variance of each one of the variables. The R squared expression, since this is correlation squared. If it's close to one, that means very good linear fit. If it's one, it means perfect in your fit. If it's close to one, it means pretty good linear fit. if it's close to zero, it's the opposite, meaning it's not good. And it's important to understand, though, that both R squared together with the residual plot or the RSS expression. They're all very useful in seeing whether we missed some systematic pattern in the data and we want to further transformed the features to take care of that additional pattern that we missed. However, they diagnosed the fate of the model to training data not to the unobserved future data, which is really where the ultimate test is, how well the model predicts future data. So we do not want to forget that. We also want to look at predictions, residuals, and likelihood values. And maybe, a sample correlation on future values. Not on the training set. We also want to do that. But the more traditional expressions of residuals and R squared, they are with regards to the fit on the training data, that is still very useful though, and we get that very easily as a product of the training stage, without looking at additional data. As I mentioned, R squared has the advantage though it's very interpretable, goes between zero and one, where one is a perfect linear fit with no noise, and zero is no correlation whatsoever between the original values and the predicted values.

For the rest of this lesson, Professor Lebanon will be using the ‘R’ programming language with the ‘diamonds’ dataset.
It is recommended the student follow along with Professor Lebanon.

To load the dataset ‘diamonds’:

```
library(ggplot2)
data(diamonds)
```

From 24 段：We're going to work in a second with the diamonds data which holds different measurements of diamonds. Every diamond is a separate instance or data point. The measurements or features are things like carat, color, cut, and the response variable that we're trying to predict in this case is price, so the price of the diamond.

Diamond prices {ggplot2}

Prices of 50,000 round cut diamonds

`Package:  ggplot2`
Version: 0.8.9

Description

A dataset containing the prices and other attributes of almost 54,000 diamonds. The variables are as follows:

- price. price in US dollars (\$326--\$18,823)
- carat. weight of the diamond (0.2--5.01)
- cut. quality of the cut (Fair, Good, Very Good, Premium, Ideal)
- colour. diamond colour, from J (worst) to D (best)
- clarity. a measurement of how clear the diamond is (I1 (worst), SI1, SI2, VS1, VS2, VVS1, VVS2, IF (best))
- x. length in mm (0--10.74)
- y. width in mm (0--58.9)
- z. depth in mm (0--31.8)
- depth. total depth percentage = z / mean(x, y) = 2 * z / (x + y) (43--79)
- table. width of top of diamond relative to widest point (43--95)

Usage

```
data(diamonds)
```

Documentation reproduced from package ggplot2, version 0.8.9. License: GPL-2

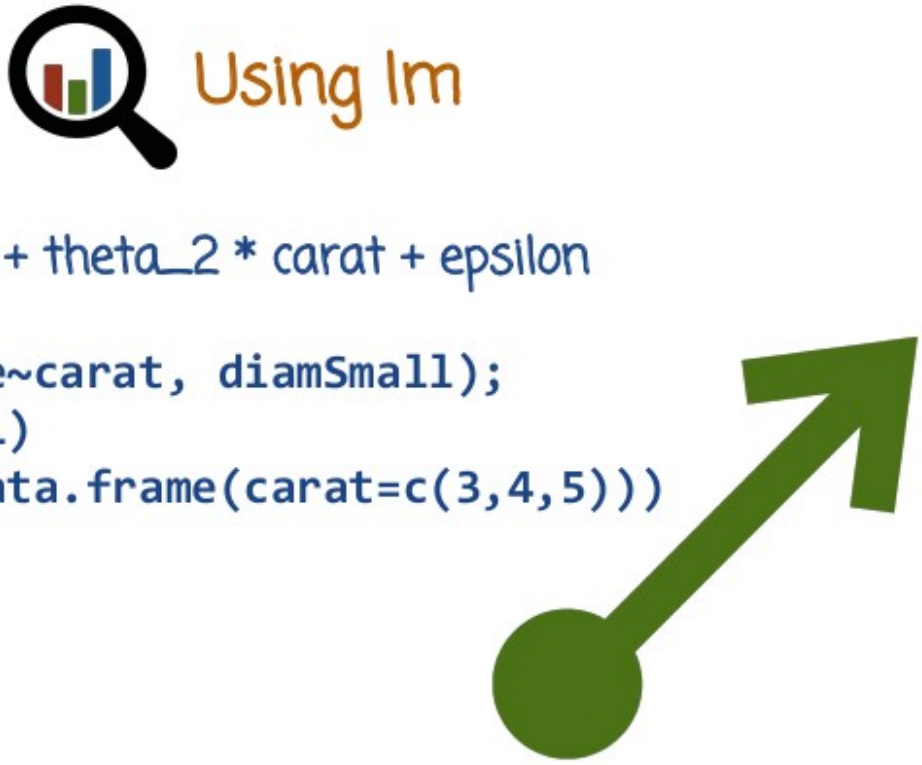
The following video nodes will use the ‘lm’ command.
You can read about this command at:
lm command

https://stat.ethz.ch/R-manual/R-devel/library/stats/html/lm.html

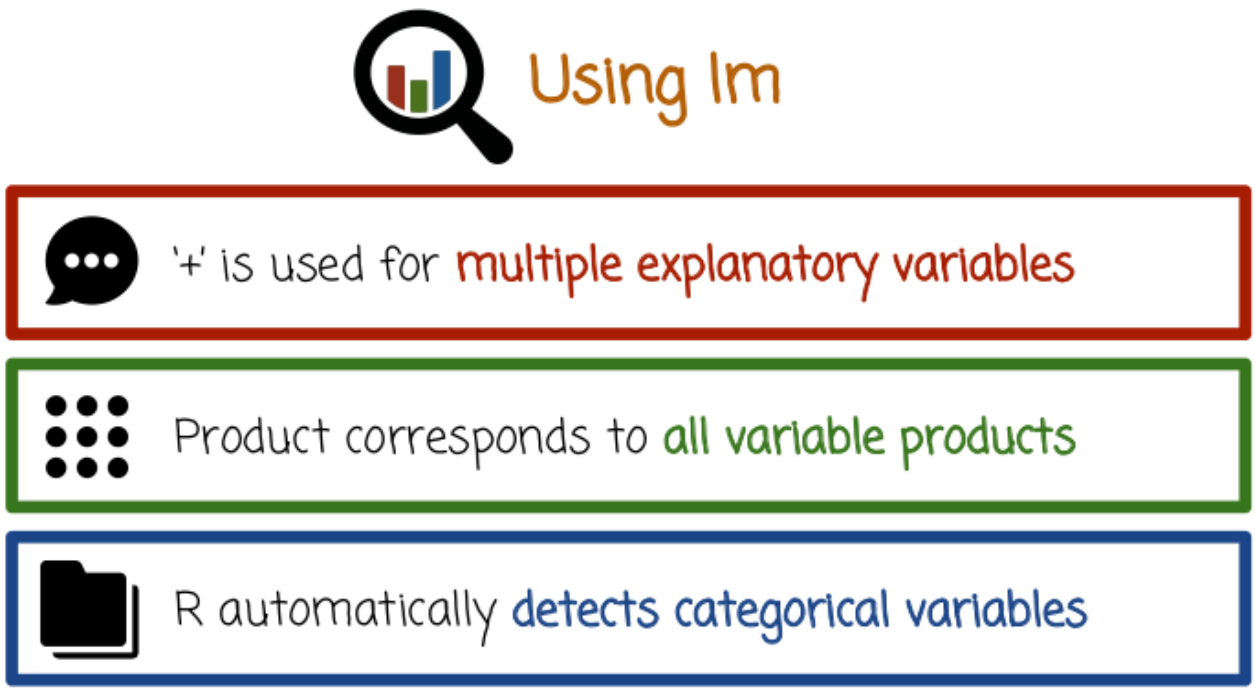
Linear Regression in R

lm(linear model formula, data frame) → M: object that can be queried
predict(M, data frame without labels) → prediction
coef(M) → get model parameters

23. So we'll see how to do linear regression in R and experience a little bit, but before we get to the actual R exercise and then the data set we'll use, let's see, some basics of linear regression in R. What do the commands look like? So the main command is called lm for linear model. And lm, the function lm takes two arguments. The first argument is a formula, a linear model formula. We'll see soon what I mean by that. And it takes a data frame which holds the training data, both the x and the y. So the x matrix and the y vector together stored in a data frame. And what it produces, the function ln returns as a result is an object that holds the linear regression model and it can be queried in different ways. One way to query it using the function predict. We pass to predict the model that we get from lm together with additional x vectors or future vectors that we want to predict based on. Then we get predictions that the model make on these vectors. Another way we can query that object is by applying the function coef, which returns the model parameters or coefficients. This is the vector theta. And there are other functions as well that can apply to the object return by lm. One additional important function is summary, which gives us a summary of the model and its fit, including things like residuals and our square value from the training process .



24. So let's see in a little bit more details. We're going to work in a second with the diamonds data which holds different measurements of diamonds. Every diamond is a separate instance or data point. The measurements or features are things like carat, color, cut, and the response variable that we're trying to predict in this case is price, so the price of the diamond. So let's just for the sake of this slide assume that the regression model is that Price = theta_1 which is the intercept form + theta_2 * carat + epsilon. So this is a one dimensional regression. There's a single x measurement in this case: carat. Epsilon represents the noise. If we want to build this regression model in R. We use the function lm with the following formula, price~carat followed by the data frame. We'll see diamSmall data set in a little bit, but for now let's just assume this is a data frame containing at least two columns. One called price and the other one called carat and when lm sees this formula right here it automatically interprets price as the response variable or y and carat as the explanatory variable or measurement or x. So this is the linear regression model and we can query, we can get the state of vector. In this case, the state of vector will have two values, theta_1, theta_2. And we can use it to predict based on the model, the price of future diamonds. In this case we pass to it the model and the data frame of future diamonds. And carat 3, 4, and 5, and that will give us the price prediction for these diamonds. And we can also display a summary of that model. And we'll see in a following video exactly how we do that in our and what are the outcomes of this exercise as well as other exercises with more complex model formulas.



We saw one simple example of a formula in the preview slide that used the notation + (要到 28 段才能看到例子, 即: M2 = lm(price~carat+I(carat^2)+I(carat^3)_diamSmall)). In general, we can keep adding more explanatory variables before in the previous formula we just had carat. But we can say carat and then add, carat plus color or cut or any other measurement variable that we have. In which case we basically just add more and more measurements or dimensions of x to the regression model. In the formula that we're using the lm function. Like I just said plus corresponds to just adding additional features to the x. If we have multiplication symbol, what this corresponds to is taking all possible products between the different features on 'the left and the right' of the product symbol and featurize that. So we'll see that later on. This is one useful way to capture all possible interactions between two vectors, and every interaction term will be converted to a measurement variable and be added to the vector x in the regression model. R will automatically detect if a measurement or a component of x is categorical, in which case it will transform it to the vector x in one of the ways we've seen previously. For example, making it into binary or creating the binary subvector that is going to be one in one component and zero in the other component based on which value is correct. So there's no need to do that manipulation explicitly. We can just keep the data in a nice format that holds a categorical variables or numerical variables mixes them up. And we don't need to do that extra step ourselves of creating an explicit numeric vector x that will be used in the training process or in the prediction process.

RStudio

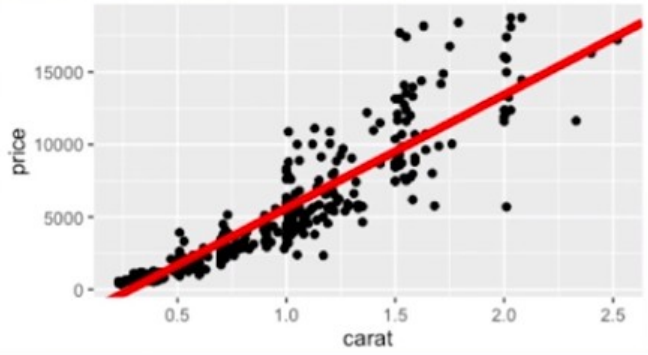
Go to file/function | Addins | Project: (None)

Console ~/

```
>
>
>
> library(ggplot2)
> diamSmall = diamonds[sample(1:dim(diamonds)[1],500),
]
> M1 = lm(price~carat,diamSmall)
> theta = coef(M1)
> theta
(Intercept)       carat
  -2240.027    7843.150
> ggplot(diamSmall,aes(carat,price)) + geom_point()
> ggplot(diamSmall,aes(carat,price)) + geom_point() +
geom_abline(intercept=theta[1],slope=theta[2],size=2,c
olor=I("red"))
> predict(M1,data.frame(carat=c(3,4,5)))
       1        2        3
21289.42 29132.57 36975.72
>
```

Environment | History | To Console | To Source

Files | Plots | Packages | Help | Viewer | Zoom | Export



RStudio

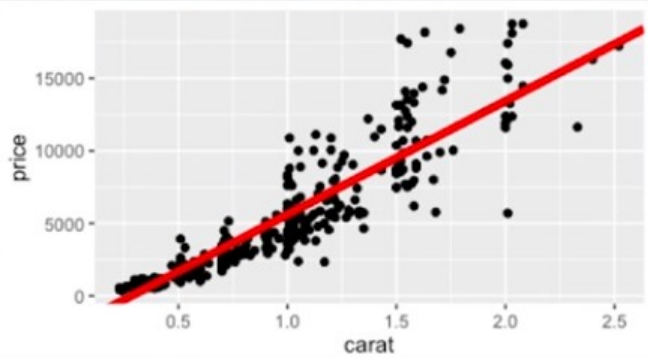
Go to file/function | Addins | Project: (None)

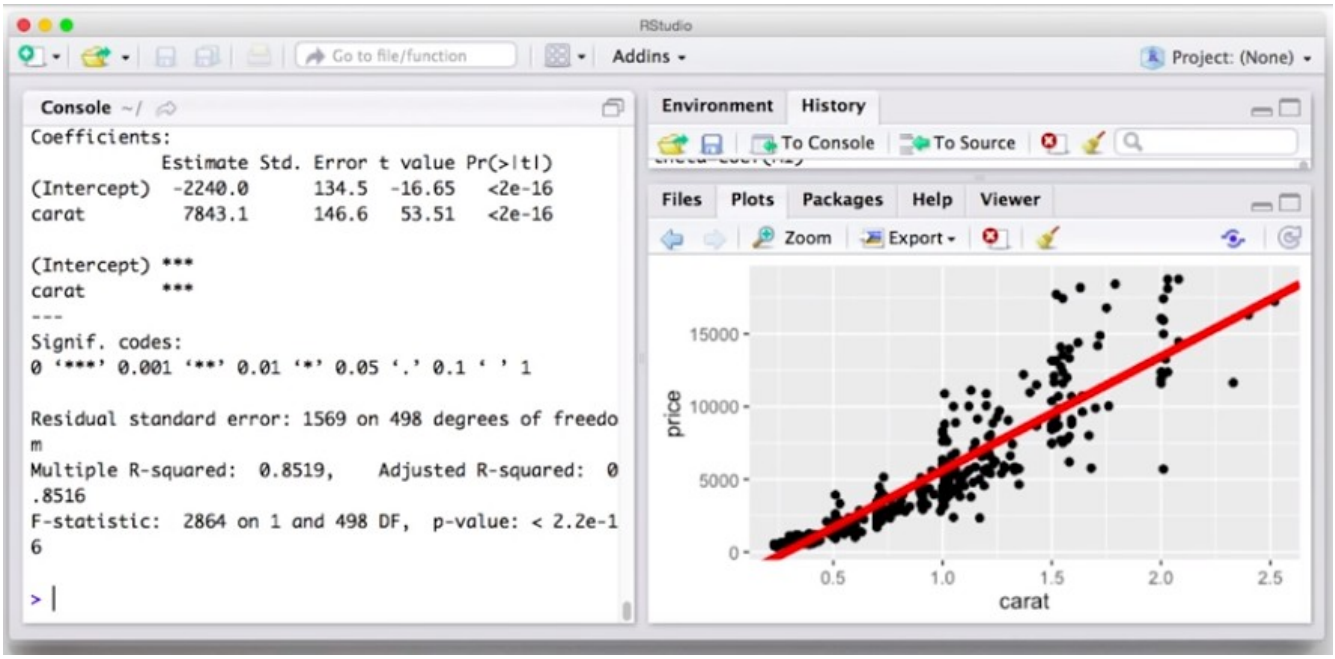
Console ~/

```
> summary(M1)

Call:
lm(formula = price ~ carat, data = diamSmall)

Residuals:
    Min      1Q  Median      3Q     Max
-7828.7  -809.3   -25.2   508.0  8041.4

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -2240.0      134.5  -16.65   <2e-16
carat         7843.1      146.6   53.51   <2e-16

(Intercept) ***
carat       ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

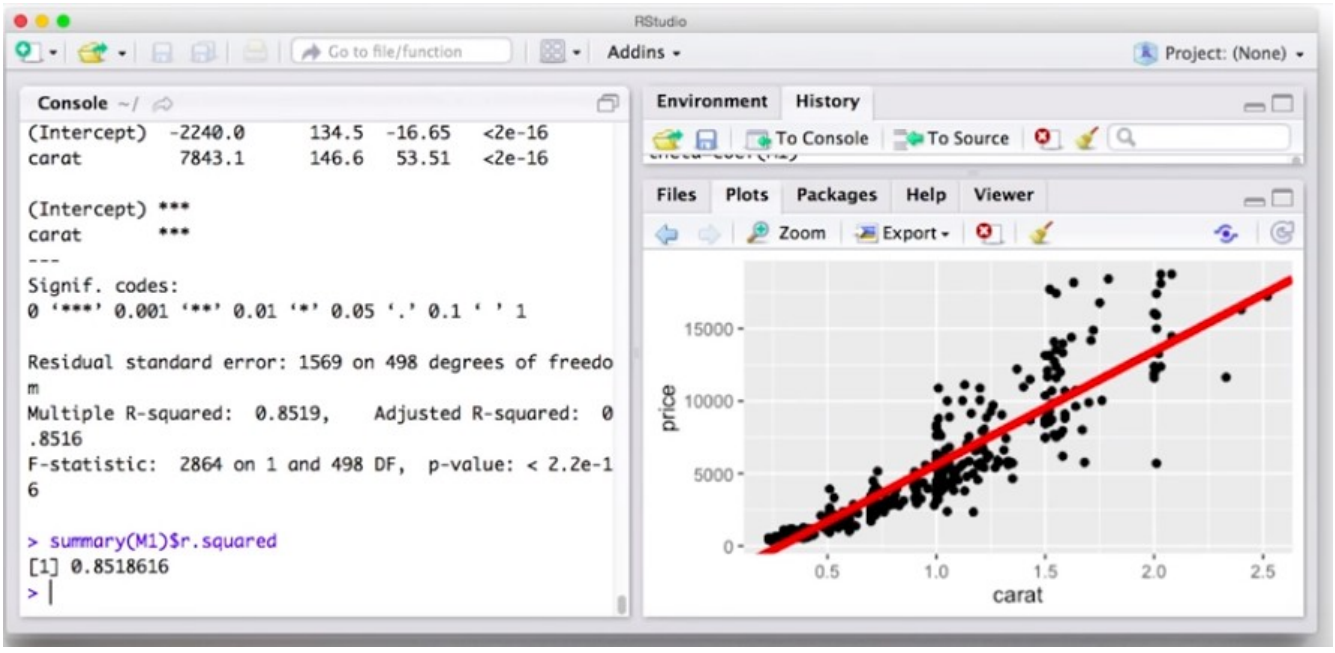
Environment | History | To Console | To Source

Files | Plots | Packages | Help | Viewer | Zoom | Export

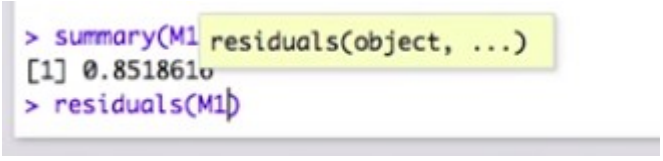


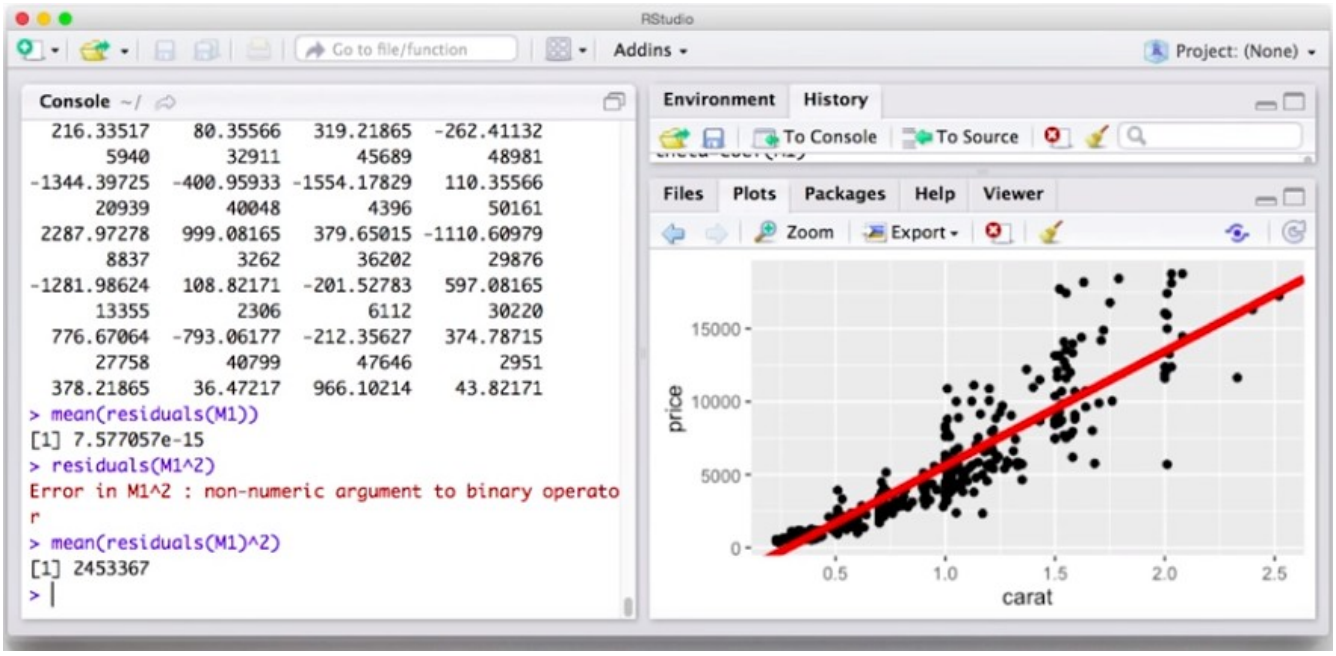

RStudio
Go to file/function
Addins
Project: (None)
Console ~/
Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept) -2240.0 134.5 -16.65 <2e-16
carat 7843.1 146.6 53.51 <2e-16
(Intercept) ***
carat ***

Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 1569 on 498 degrees of freedo
m
Multiple R-squared: 0.8519, Adjusted R-squared: 0
.8516
F-statistic: 2864 on 1 and 498 DF, p-value: < 2.2e-1
6
>
Environment
History
To Console
To Source
Files
Plots
Packages
Help
Viewer
Zoom
Export
price
carat
15000
10000
5000
0
0.5
1.0
1.5
2.0
2.5


RStudio
Go to file/function
Addins
Project: (None)
Console ~/
(Intercept) -2240.0 134.5 -16.65 <2e-16
carat 7843.1 146.6 53.51 <2e-16
(Intercept) ***
carat ***

Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 1569 on 498 degrees of freedo
m
Multiple R-squared: 0.8519, Adjusted R-squared: 0
.8516
F-statistic: 2864 on 1 and 498 DF, p-value: < 2.2e-1
6
> summary(M1)$r.squared
[1] 0.8518616
>
Environment
History
To Console
To Source
Files
Plots
Packages
Help
Viewer
Zoom
Export
price
carat
15000
10000
5000
0
0.5
1.0
1.5
2.0
2.5

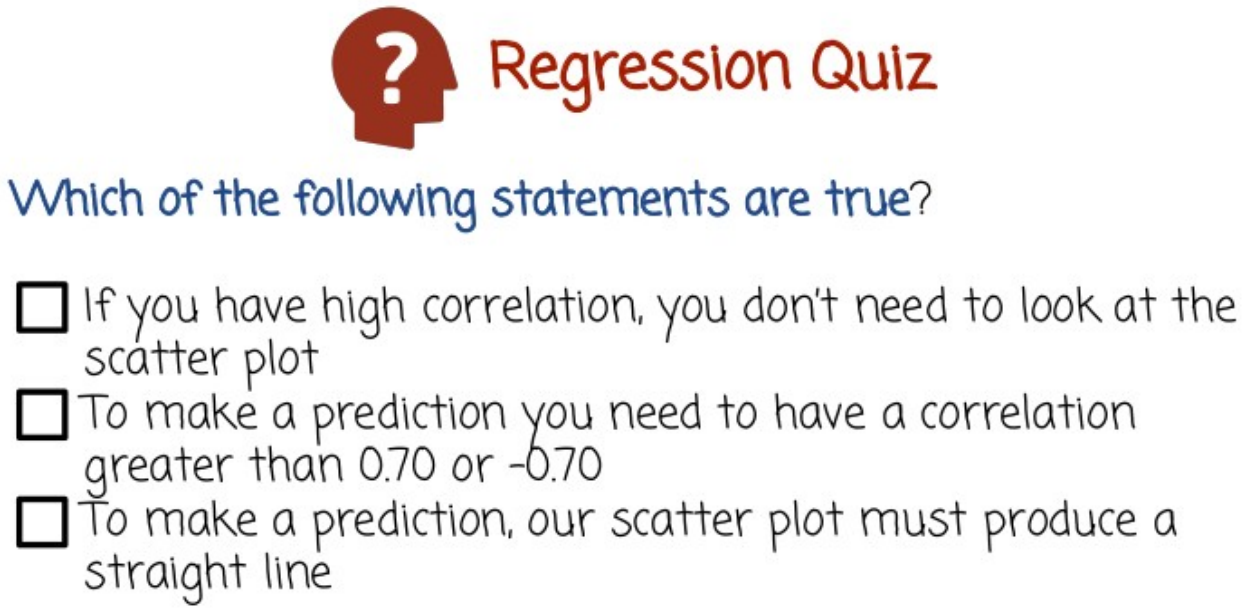

> summary(M1
residuals(object, ...)
[1] 0.851861
> residuals(M1)



25. So let's see how we build the simpler regression model in R. We're going to use for visualization ggplot2. So if it's not yet in scope, we'll first need to be installed. But if it's installed, we first need to bring it in scope by using the library function. Then what we need to do next is create random subset of the diamonds data frame which is included in the ggplot2 package. And we're going to sample 500 rows randomly and we want it to be random. So, what I'm doing here is I'm creating first a sample of 500 row indices between one and the number of rows in diamonds, which is dim(diamonds)[1]. And then I extract these rows from diamonds data frame to create a new data frame diamSmall with these 500 random rows. Okay, so let's build the simple regression model which we going to call M1. And in this case, we're just going to model price as a linear function of carat. And the data frame is damn small. Now if we want to look at the coefficients that we get in this model, we can use the coef function. So we have two coefficients. The intercept, which is theta one, multiplying basically by one. Or it doesn't multiplied by any explanatory variable or explanatory variable equivalent to one. And then theta 2, which multiplies carat. Okay, so let's try to visualize this. So, we first need to call ggplot with the right dataframe and statics and then we're going to add scatter plot. We see here on the right hand side scatter plot of price versus carat for the 500 randomly selected diamonds. But we also want to add now the model line, so we see how it fits on top. So we need to add another layer. And we're going to use the geom_abline, which is basically drawing a line using A and B coefficients or intercept in slope formula. And so we're going to say intercept equals theta 1. And slope equals theta 2. And the size of the line will be 2. So it's a little bit wider than normal. And the color may be different colors, so it's a little bit more visible. And we see here, let me make it a little bit bigger, that the regression line that we learn, it looks like it's a pretty good predictor of the relationship between price and carat. We can use that to predict future values. We'll see later ways on further improve upon that, okay. But let's see now what else we can do with the model besides just plotting it. So we can, for example predict new values, so we're going to call predict using the model, that we got back from lm, and with the data frame that we're going to create, that is going to hold carat values that we encounter in the future. And we get back the prediction of the model that we just learned on these three different, three newly observed carat values. Notice that the carat values were up here predicting 3, 4, and 5 are quite high and our training set here is only up to 2.5 carats. So it's kind of like extrapolating a little bit. We don't

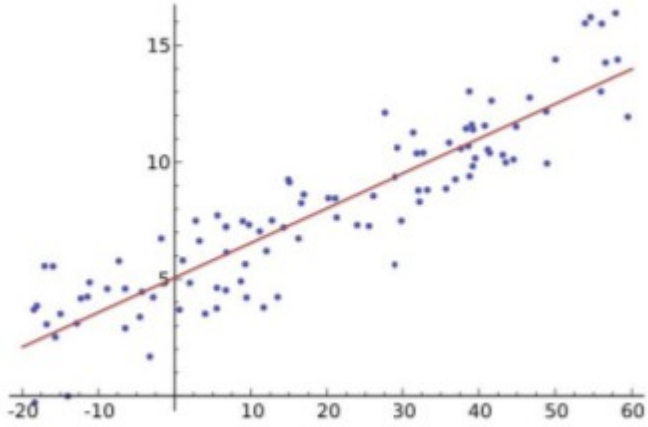
see these large values, carat 3, 4 and 5, in our training sets. So it's quite possible, in this range, the linear relationship will no longer hold. So the straight line may not be a good description of price in these higher carats. But nevertheless, we can use the predict function to do just that. Another thing we can do is we can called a summary function, which gives different properties of the model. So we can see the statistics of the residuals, the minimum, first quartile median, third quartile maximum. We can see the coefficients and we can also see the r squared coefficient. If we want to get specifically the r squared coefficient for example, what summary function returns is actually a data structure which we can refer to specific fields in it using the dollar notations. So for example, the r squared is precisely that. And if we want residuals, we can use the same thing, but we can also use the residuals function. And we want the average of the residuals, we just assay to the function mean or we can just do the sum or the summation. Notice actually we probably want the average of the residual squares. Because when you just take the average of the residuals here, the very, very negative errors cancel out with the very, very positive errors, you get something very small. But if you actually get the residuals squared, you get to basically accumulate the squares of the error so that you don't cancel a very big error positive error with a very big negative error. So you could do mean or you can do sum the RSS is the sum, the mean is the same as the sum but divided by the number of residuals.

26. Which of the following statements are true? If you have high correlation, you don't need to look at the scatter plot. To make a prediction you need to have a correlation greater than 0.7 or -0.7. To make a prediction, our scatter plot must produce a straight line.

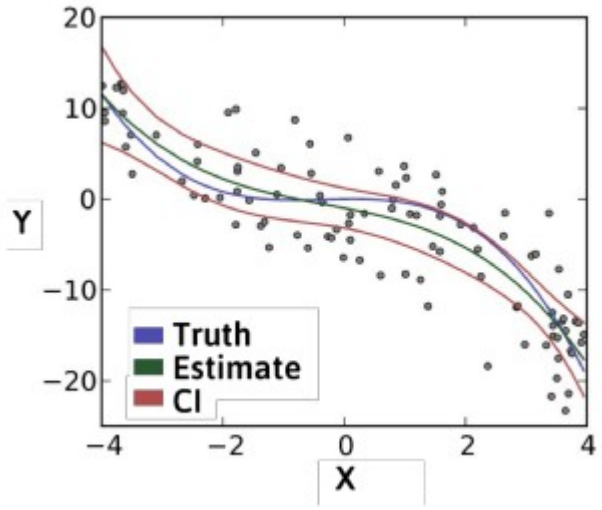


27. The first statement is false. It's very useful to look at scatter plot even if looks like the correlation or r squared is pretty high. Scatter plot of the data set and the regression line or residuals, it can reveal additional information that may be useful to know whether you want to revise the model or not, even if you have high correlation. The second statement is also false, there's no such rule. It's quite possible that the correlation between y and y hat is lower than 0.7. It could be because of the fact that the model is not a good model, the linear model that you used is not a good model. Or it could be because of the fact that there's just a lot of inherent noise in the data. In any case, you can still make prediction given by the linear regression model. You just need to associate with that prediction the uncertainty in the confidence you have in that prediction. If that's the only model that you have, quite possibly some

prediction is better than no prediction, you just want to be careful not to mislead anyone, and not to have a high confidence in the result when the data or the model shows that there's actually a pretty high noise In the data and in the prediction. The last statement is also false. You can make a prediction if the scatter plot does not produce a straight line. For example it could be that you're using non-linear transformation and the relationship between the x and the y are not a straight line and also the regression line in the original space is not a straight line, but in the transformed space it is a straight line. And you may still make the prediction, like I said, the same caveat that I mentioned earlier still holds, you need to associate the prediction with some confidence value or some statement to the extent that this is an accurate prediction or a rough approximation.

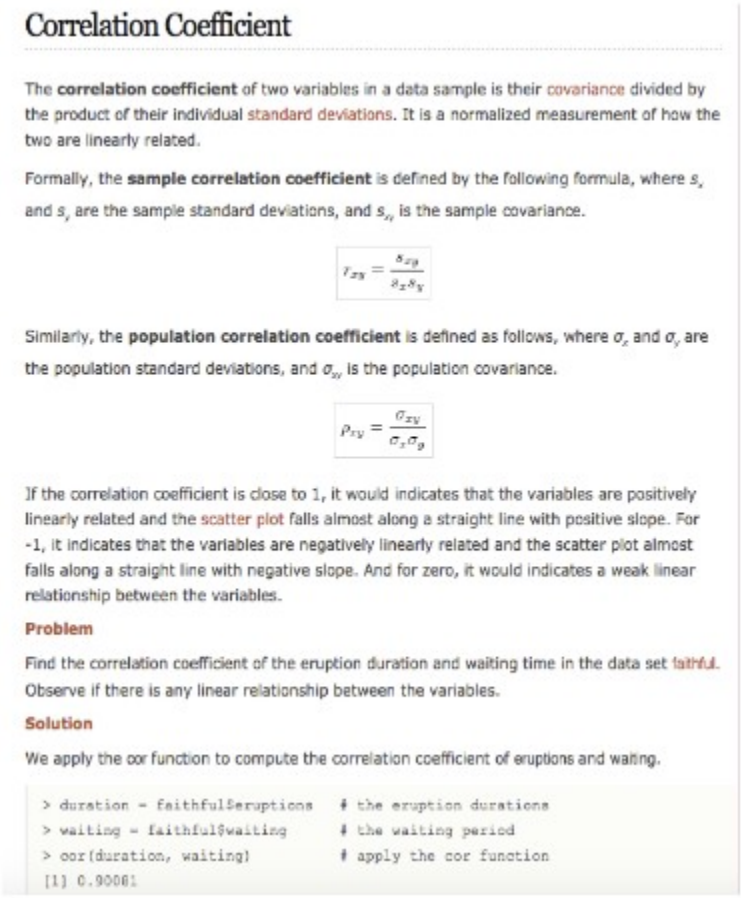


So here's an example of pretty nice linear relationship between x and y with a pretty nice model depicted by the red line right here. And you can certainly make a prediction here for future values but certainly It makes sense also to look at the residuals the R squared and talk about the amount of noise that you have in the prediction.



And here is an example of non linear regression in the original space given by linear regression on transformed values.

The cor command is used in 'R' to calculate the correlation coefficient.



Correlation Coefficient

The **correlation coefficient** of two variables in a data sample is their covariance divided by the product of their individual standard deviations. It is a normalized measurement of how the two are linearly related.

Formally, the **sample correlation coefficient** is defined by the following formula, where s_x and s_y are the sample standard deviations, and s_{xy} is the sample covariance.

$$r_{xy} = \frac{s_{xy}}{s_x s_y}$$

Similarly, the **population correlation coefficient** is defined as follows, where σ_x and σ_y are the population standard deviations, and σ_{xy} is the population covariance.

$$\rho_{xy} = \frac{\sigma_{xy}}{\sigma_x \sigma_y}$$

If the correlation coefficient is close to 1, it would indicates that the variables are positively linearly related and the scatter plot falls almost along a straight line with positive slope. For -1, it indicates that the variables are negatively linearly related and the scatter plot almost falls along a straight line with negative slope. And for zero, it would indicates a weak linear relationship between the variables.

Problem

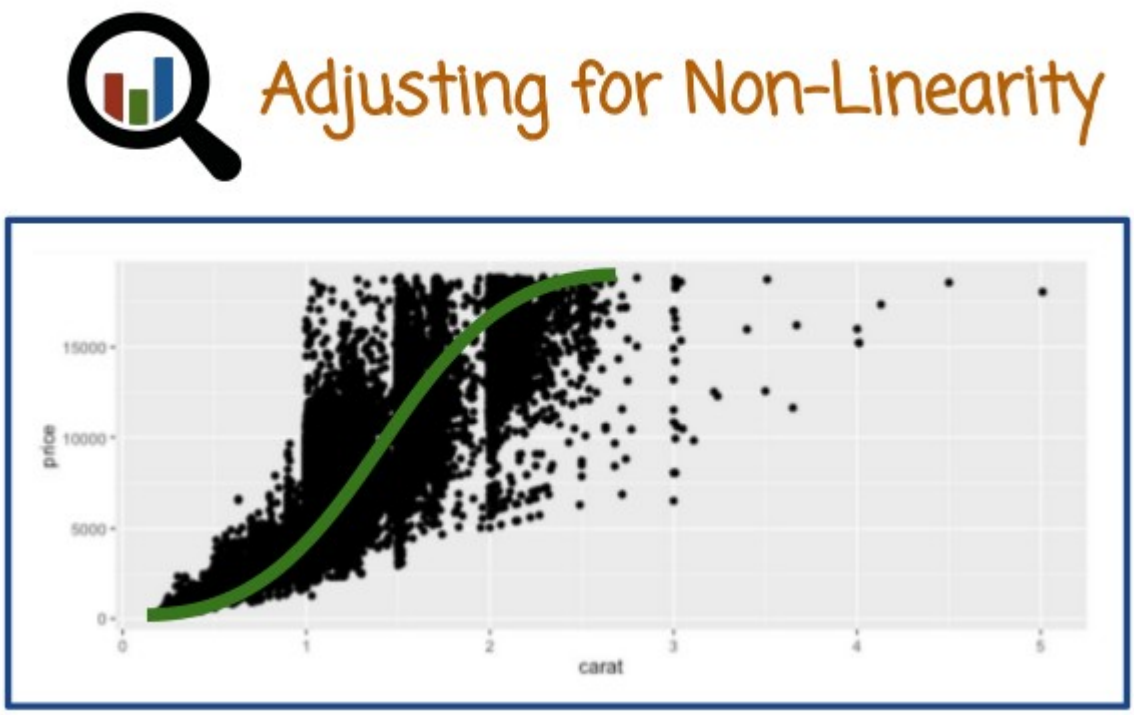
Find the correlation coefficient of the eruption duration and waiting time in the data set faithful. Observe if there is any linear relationship between the variables.

Solution

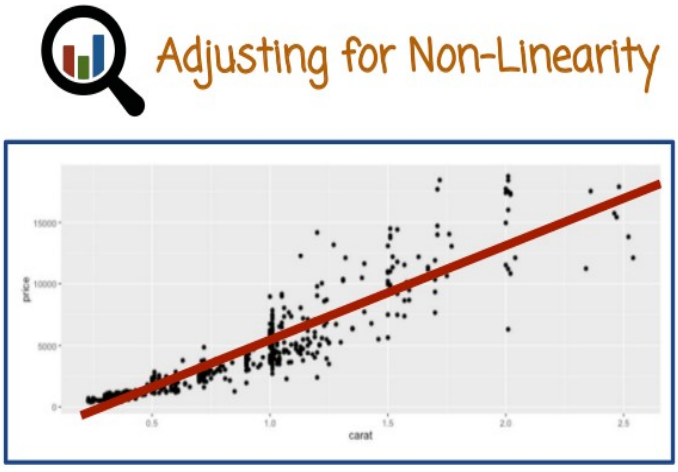
We apply the cor function to compute the correlation coefficient of eruptions and waiting.

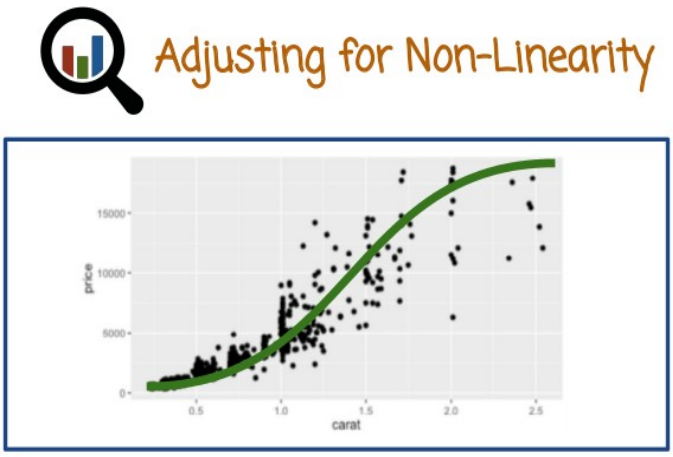
```
> duration = faithful$eruptions   # the eruption durations
> waiting = faithful$waiting      # the waiting period
> cor(duration, waiting)          # apply the cor function
[1] 0.90081
```

上圖原圖就這麼大. 注意上圖中的 correlation coefficient 跟前面的 R^2 不同, R^2 不能為負, 而 correlation coefficient 可以為負.



28. What we have here are a scatter plot of the price versus carat for the diamond's dataset. And it looks like intuitively, or just by looking at the data, there's certainly an increasing relationship between price and carat, but maybe one straight line is not necessarily the best fit. Maybe something like an S-shaped curve here will be a better fit for the data, and would give us better prediction for future diamonds.





This(左) is the linear regression model that we got in the previous programming node. It is not a bad model, as you see, it captures the signal in the model, to some extent. But as we saw before, when you look more carefully at the data (右), it looks like maybe a nonlinear shape would be a better model. So let's think how do we go about doing that in a little bit more detail followed by an R exercise.

Adjusting for Non-Linearity

```
Price = theta_1 + theta_2*carat + theta_3*carat^2
+ theta_3*carat^3 + epsilon

M2=lm(price~carat+I(carat^2)+I(carat^3),
diamSmall);
```

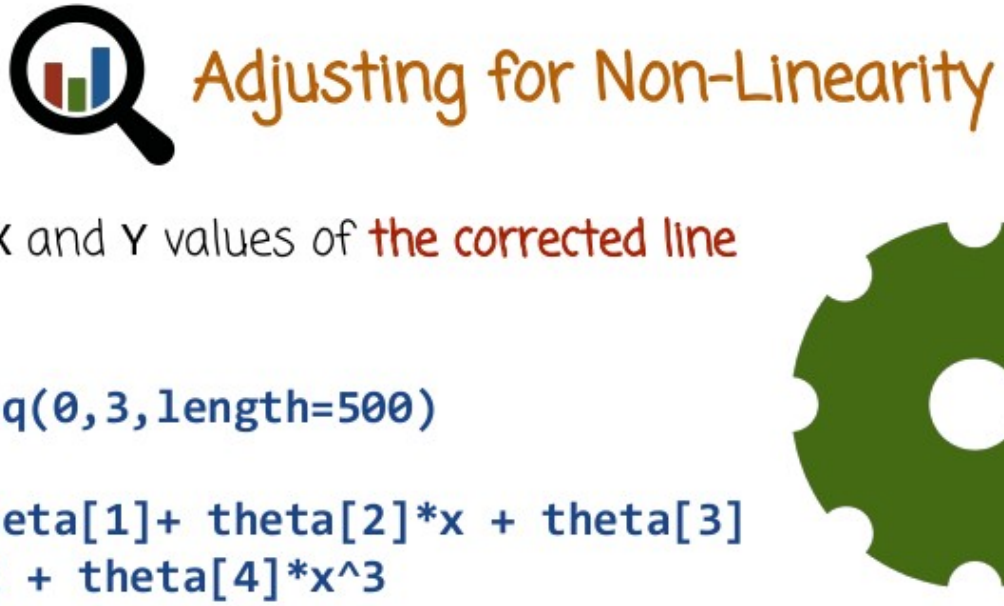
So one way to do that is by just taking a nonlinear transformation of the original values so for example, we can use the following regression equation. Price is theta_1 which is the intercept, + theta_2*carat. So if we stop here, we get back the linear regression model from the previous video which we refer to as m1. But now, we're going to also theta_3*carat squared, and theta_4*carat to the third power. And then of course we need to add the epsilon term, which is the noise. So this is the regression model assumption, or formula. This is not the formula that we use in the LM functioning R, this is just the mathematical expression of the linear regression assumption we're going to try to use. The R code that we would use to build that regression model is we're going to use the following formula, lm(price~carat+I(carat to the second power)+I(carat to the third power). And remember, the plus sign can be used to add more explanatory variables. And then we need to indicate the data frame name that we're going to work with.

Adjusting for Non-Linearity

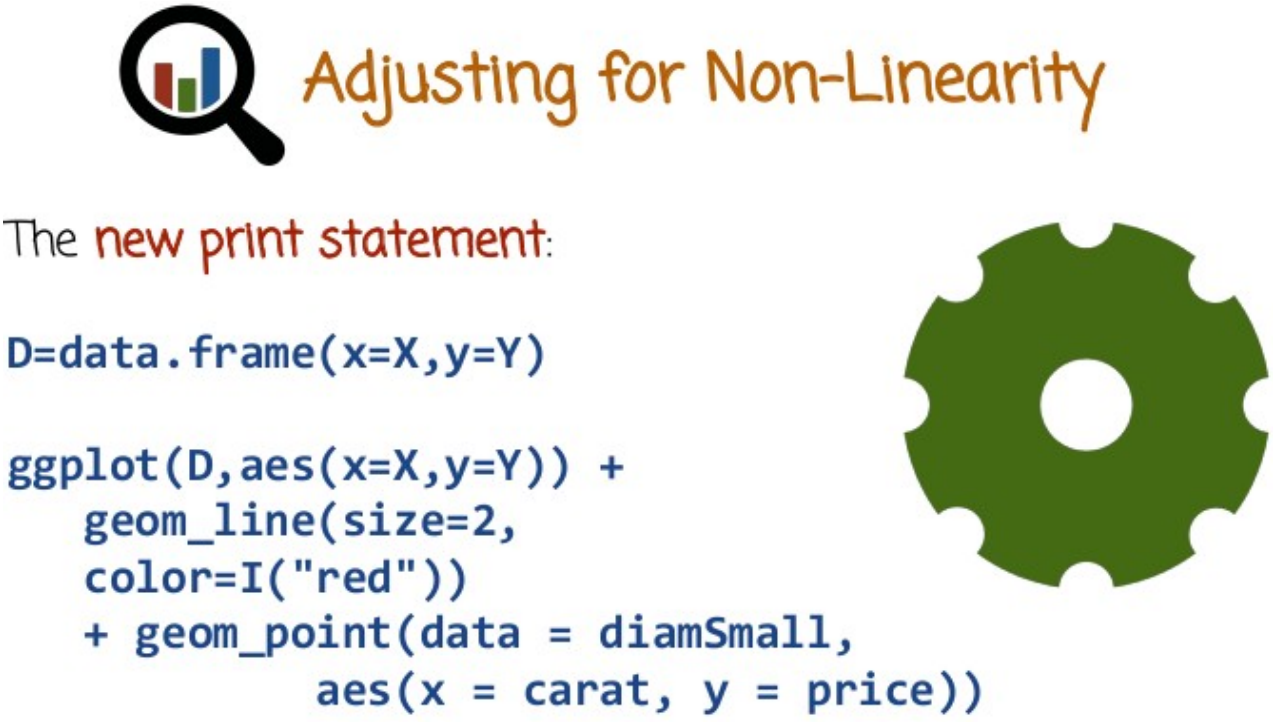
The least squares parameter estimates

```
theta = coef(M2)
```

After training we can, for example, query the model m2 using the function coef, which returns to us the vector theta, the vector of parameters, in this case theta would have four different components. We can also query them using the summary function or predict if we want to predict future values.



29. So, continuing the example from the previous video we'll see how to visualize the new non-linear model and two that we build which is linear in the transformed variables but non-linear in the original variables. So first we create a grid of values for x axis, 500 equally spaced points between zero and three. Then we create a new vector y with the predicted values given by the regression equation here applied to the vector x of equally spaced points with the theta that we got from the model that we just trained M2.



Here's how we can visualize the new regression model. We first create the new data frame with the

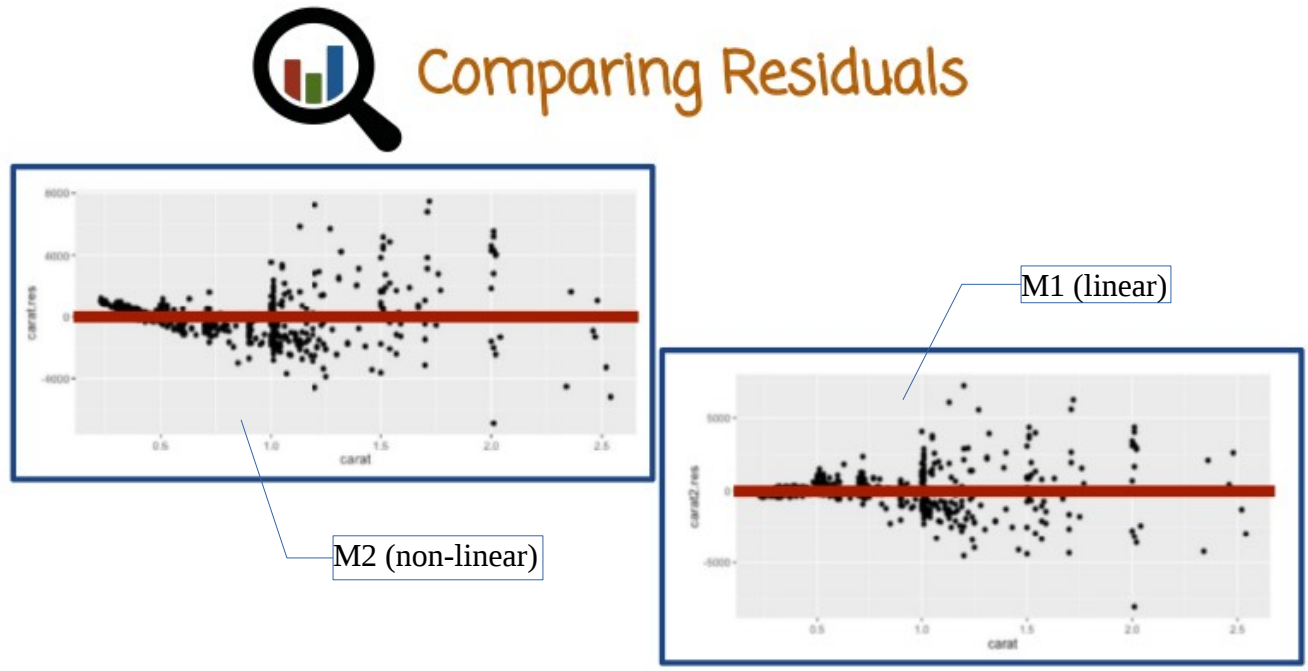
data that we just created, and we need to have two components here, two layers in the GGPlot function. The first one is going to be a line that shows the regression line. Now that's going to be non-linear because we're going to plot the N2 model in the original space of just carat. In that space it will be non-linear based on this data frame here. It's going to be slightly thicker line and color red and overlaid with that we want to add the points in the training data so we see what the fit looks like.

30. So let's see now how we can build a new model for price based on carat by introducing nonlinear transformations to carat and building a regression model that is nonlinear in the original variables, in this case variable carat, that hopefully will be a better fit. So we build a new model for price as a function of carat. But we also add here carat to the power of 2 plus carat to the power of 3. And maybe that's enough, but we need to also pass the theta frame. And let's look at the coefficients. So, we see in

this case, previously we saw only two coefficients, the intercept and the carat coefficient. Now, we have four coefficients because the regression is done on a four dimensional x vector where x1 is identically 1, x2 is carat, x3 is carat squared, and x4 is carat cubed. Okay, so now let's visualize the new model and overlay it with the scatterplot of the datasets. We need to create a grid of 100, let's say, equally spaced points between 0 and 3. That will form the grid. So we can no longer use AB line, which we did before to get this line. Now we need to use a different layer called geom line. And you need to manually create the grid and have a vector of values corresponding to the y coordinate of that nonlinear function so that we can properly display it. So first we need to create the grid of 100 equally spaced points. And next we need to evaluate the regression model on this grid. So we have theta[1], this is the intercept, +theta[2]*X+theta[3]*X^2+theta[4] *X^3. Okay, now we're going to create a dataframe with these values. going to call ggplot with that dataframe and we're going to add to it the line geometry in order to plot the regression line. But we're going to plot it in the original space, in the original one dimensional space, and there it will be appear to be non-linear. Now it's going to be hard to visualize in the transform space because it's a four dimensional space. But let's see what it looks like in the original space. So let's make it a little bit bigger and with a red color. And so, this is the line that the regression returned, the line in the four dimensional space now looks like a curve in the one dimensional space X. So let's overlay it together with the points corresponding to the dataset. So we need to add a point geometry with a different data frame, diamSmall. And we need to tell it what will be the x and what will be the y. And here it is, the non-linear regression, which is linear in the four dimensional space and overlaid with the scatter plot. And now we see that we do capture here the fact that for higher carats it seems like at the beginning it looks like the trend is not quite as steep as the linear regression picked it up. The first simple linear regression we tried first time, one. So here, the slope is lower and then the slope increases. But here, it looks like the slope again is starting to decrease. Now this part here, I would not believe it too much for two reasons, the downward slope. The first reason being that we don't have any points beyond that value here. So it's difficult to make prediction in areas, or regions where we don't have near by training points, but the other reason is that we know that usually a price increases with carat. And it seems kind of suspicious that larger diamonds would be cheaper. So that's probably an artifact right here of the fact that we don't have data in this region to guide the model to learn that the increasing trend continues. Now if we want to like before explore the model, use it for prediction and so on, we can do it. So we can, for example, look at the r squared coefficients. In this case, it's higher than the r squared of the previous model, which seem to indicate this is a higher correlation between the ground truth and the predicted values which seem to indicate a better prediction. And we can also look at the average of the squared residuals. And in this case, this value right here, let's see what we got before. Before we got a much higher value of R assess. Well, this is the mean, so that's normalized, but the unnormalized R assess would have the same trend. So the previous mean of the squared residuals, or also, sum of squared residual was quite a bit higher than the new one, indicating that the new model using the non-linear transformations is a better fit.

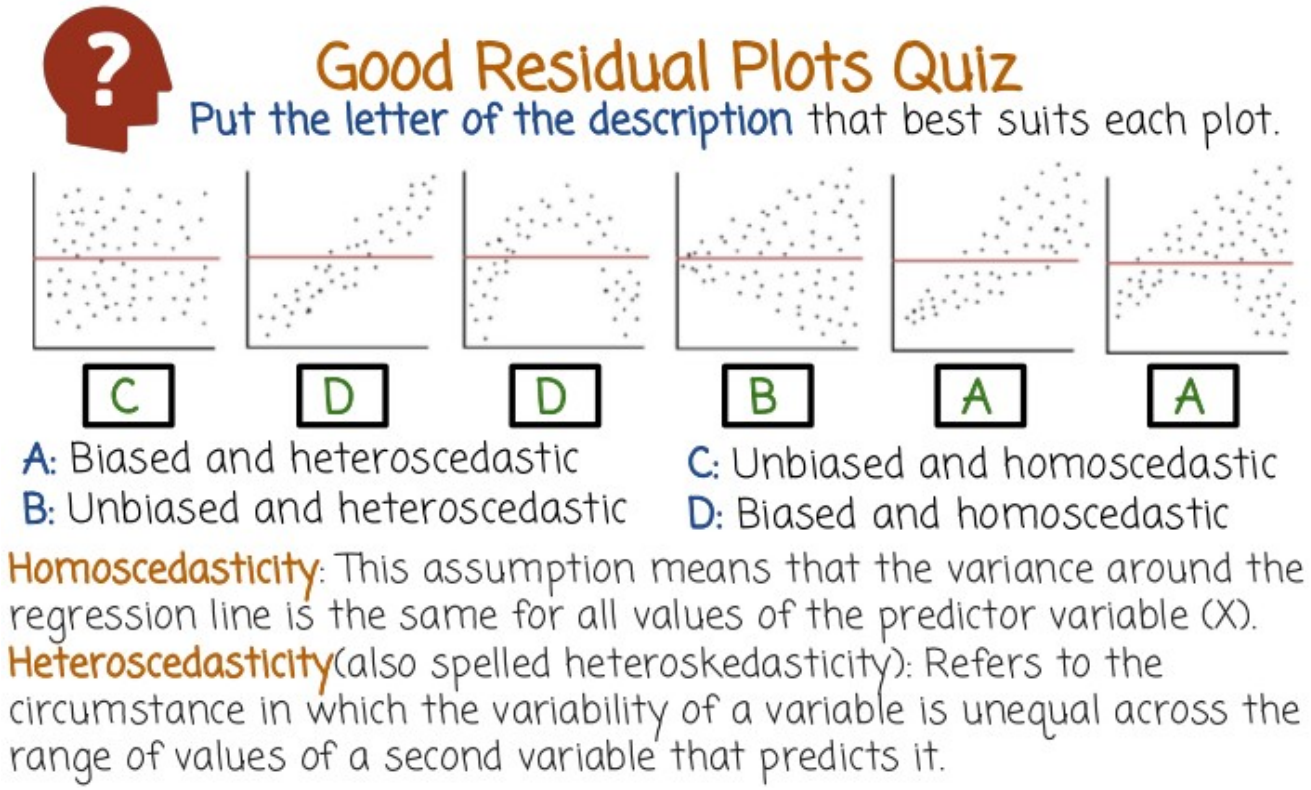
	M1 (linear)	M2 (nonlinear)
R^2	0.840053	0.855261
RSS	2618097	2189109

31. So, if we want to compare the two models that we tried previously, M1 which is linear just using carat, and M2 which is nonlinear in carat because it uses additional powers when constructing the linear regression model. One way to do that is to look at R squared and residual sum of squares for the two models. And if you do that yourselves, by the way, you may get slightly different values, because this based on a random subset of the diamonds data. And if you do the random sampling yourselves, you may get slightly different samples. But we can compare the two models here, and you can do that as well with your own sample. And it looks like the R squared coefficient or squared correlation coefficient is higher for M2, which means it's a better fit. Also the RSS or residual sum of squares is lower, also indicating a better fit. So we can conclude from that that it looks like the M2 model is a preferred model. It's nonlinear in carat although it's linear in the transformed space. It looks like the additional complexity of the nonlinear transformations actually helped us get a better model. Although, like I said earlier, we always want to have the caveat that this is a fit to the training data. And we do want to also check and see how the models perform on future unseen data sets.



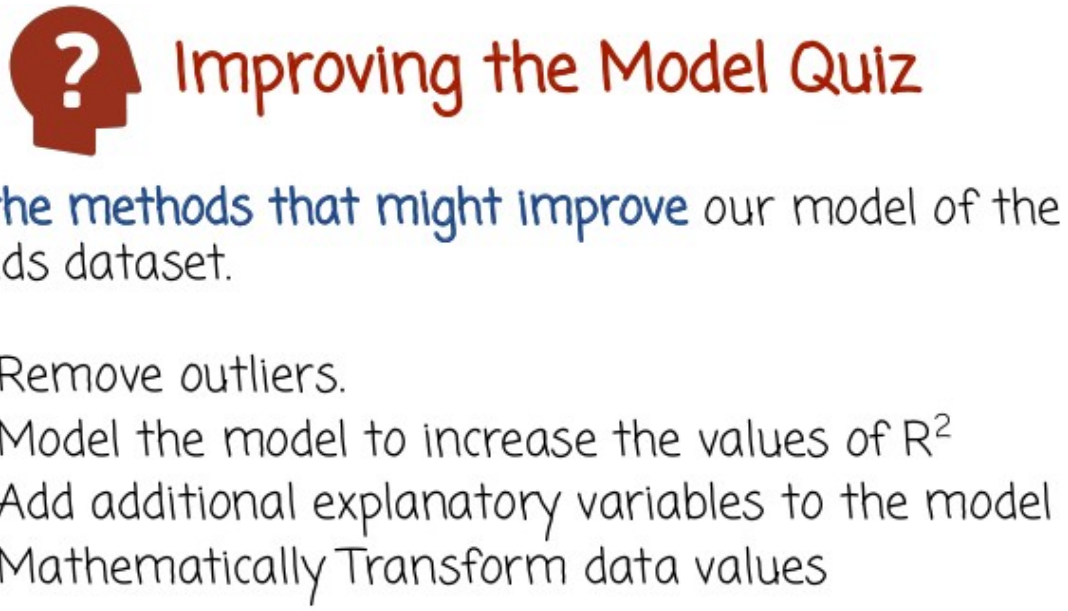
Here's the residual plots. If you recall, a residual plot is a scatter plot where the x is an index of a training example. And the y is the difference between the ground truth value of y and the predicted value of y. And so as you can see in the right hand side is the residual plot of the M1, the linear model. We do see here some systematic deviation. Specifically, it looks like the residuals first are trending down and then they trend up. And maybe then they trend down again. The residual plots for M2 look much more centered around zero, although we do see something interesting in terms of the spread of the residual. Looks like the spread is increasing, which actually makes sense. Because one could argue that higher carat diamonds have a bigger spread in price. For example, it could be perfect diamond or flawed diamond or depending on the shape or the color or whatever the price could vary a lot. So that's not necessarily something we should be concerned with but we should be aware of the fact that the residuals increase with the carat.

32. In this quiz, place the letter A, B, C, and D into the description that best suits each plot. So A, biased and heteroscedastic, B, unbiased and heteroscedastic C, unbiased and homoscedastic, D, biased and homoscedastic. We haven't seen these concepts before. Homoscacedicity assumes that the variance around the regression line is the same for all values of the predictor variable x. And heteroscedasticity refers to the other situation where the variability or the variance is unequal, the variance along the regression line is unequal across the ranges of the values x. So, this basically corresponds to whether the residual plots is going to be of uniform noise or randomness around the zero line.

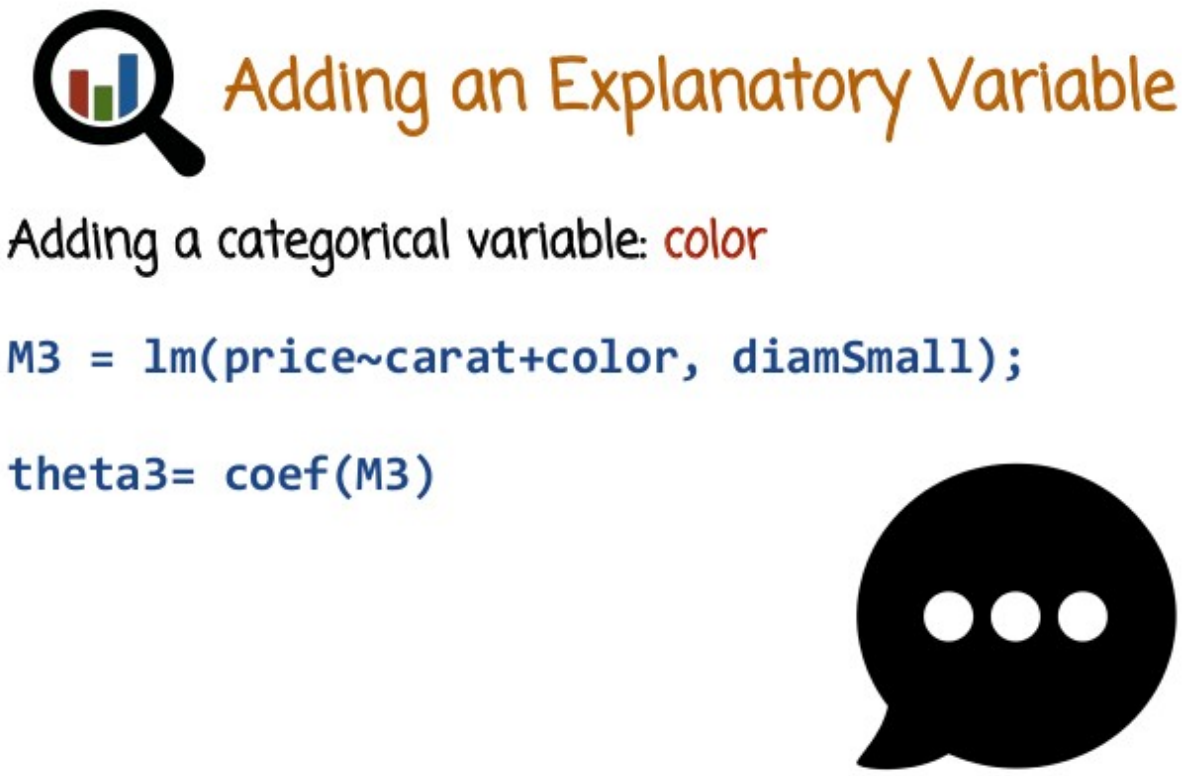


33. So in the first case, this is C, unbiased and homoscedastic. Unbiased because looks like the residuals are equally distributed along the red line (即以 red line 為對稱軸) marking the regression prediction. And homoscedastic because the distribution of the Y values is similar along different x coordinates. In this case, the letter is D, biased and homoscedastic. So it's biased because the residuals are not centered around the red line. And it's homoscedastic because the variance (數据點的 span) along the y axis is the same for different values of x. In this case, we have D as well. It's biased and homoscedastic for the same reasons. In this case it's B, unbiased and heteroscedastic. So unbiased because the points are aligned roughly equal on top and below the red line. And it's heteroscedastic because the variance opens up as you move to the right. Here you have biased and heteroscedastic. You can see the bias in the points along the red line, as well as the heteroscedasticity shown by the variance opening up as you move to the right. And the last one as well, is A for the same reasons.

34. Select the methods that might improve our model of the diamonds dataset. Remove outliers. Model the model to increase the values of R-squared. Add additional explanatory variables to the model. Mathematically transform data values.



35. The first statement here I marked as false, although one could argue that removing outliers may help, but we didn't quite see evidence of that in the diamonds data set. So, it's possible this will help but based on what we've seen so far we don't have strong reason to argue that there are extreme observations or outliers that if we remove them, we'll perform better. Not necessarily a bad thing to try, can try it out, but probably not something that we can conclude based on what we've seen so far. The second statement, model the model to increase the values of R squared. What you want to do rather than model the model is to do things like add additional variables, and that may increase the values of R squared rather than model the model. The last statement is also correct. Mathematically transform data values. That can also help out in improving the model. Could even improve R squared, could decrease RSS, and most importantly could help us predict better on future data.



36. Well, let's see a new linear regression model, M3 that is based on 2 variables, carat and color. And interestingly, color is a categorical variable. Like I said earlier R automatically detects it, and will handle it appropriately using one of the methods we described before. For example, converting the categorical variable into a binary subvector that will be concatenated to carat. The formula here just has a plus sign, which means we're going to add both explanatory variables to their regression model. If we want to get back to thetas, we can query the model M3 as we saw before. And we'll see in the future video what happens when we do that in R, and what happens to the model theta as well.

37. So we saw before two examples of linear regression models for price based on carat. But let's see now a third linear regression model that looks not just at carat or nonlinear transformations of carat. But also looks at another variable, in this case, color. So we call lm function with the formula price is a function of carat, but now add color which is a different variable or column name in the data frame. DiamSmall, and we can look at the r squared, so we compare the r squared here to this r squared, it looks a teeny bit higher. So it looks like slightly better correlation between the ground [INAUDIBLE] and predicted values. So 869 instead of 868 or M2, let's see what we get for the residuals. So the residuals in this case are lower than the residuals in M2. And so it looks like M2 is clearly better than M1. It has both lower residual, lower RSS, also has a better r squared coefficients. M3, the r squared coefficient is slightly worse, but the RSS looks like actually a nice improvement over M3, looks over M2. So it looks like the addition of the color gave us a nice additional signal.

	M1	M2	M3
R^2	0.840053	0.855261	0.8589676
RSS	2618097	2189109	2308494

38. So let's compare the three models we've looked at so far. M1 is the linear model using just carat, M2 is the non linear transformations of carat and M3 is carat plus color. And we can look the like we did earlier the R squared and the RSS although I would probably want to look also at the residual plot and we also want to look at performance on future data or held out data and so on but for this slide lets look at the R squared and RSS we see that M3 looks like it has the highest R squared. In terms of RSS, looks like M2 has the lowest RSS. And so the R squared is very close between M2 and M3. But in terms of RSS, M2 has a lower RSS, then M3. So if we want to judge them all just based on the RSS, then M2 would make more sense. It's making overall less smaller value of square errors. In terms of model correlation, M3 is slightly higher. But looks like overall the increase in squared correlation (R^2) is fairly small. In particular when we compare it to the increase correlation between M1 and M2. Pretty significant increase but from M2 and M3 very small. On the other hand nice reduction in RSS. So if we had to choose perhaps M2 would be a good choice to proceed and to look at how we perform in future data.


```
M4=lm(log(price)~log(carat),diamSmall)
```

39. Let's see one more way of a variation on logistic regression which we'll call M4. And what we'll do here is rather than predict price as a liner function of carat, we're going to predict the log price as a liner function of the log of the carat. And so we apply a transformation to both the explanatory variable x and to the response variable y. In both case we use a log transform. And R knows how to parse this equation so there is variables on both sides of the tilda sign that are column names in the data frame here, and then there is transformation applied to that name. In this case a log.

```
> summary(M3).r.squared
Error: unexpected symbol in "summary(M3).r.square
d"
> summary(M3)$r.squared
[1] 0.869746
> mean(residuals(M3)^2)
[1] 2157178
>
>
>
> M4 = lm(log(price)~log(carat),diamSmall)
> theta(coef(M4))
Error: could not find function "theta"
> theta=coef(M4)
> theta
(Intercept)  log(carat)
   8.459527    1.680302
>
```

```
> theta(coef(M4))
Error: could not find function "theta"
> theta=coef(M4)
> theta
(Intercept)  log(carat)
   8.459527    1.680302
> ggplot(diamSmall,aes(log(carat),log(price))) +g
eom_point()
> ggplot(diamSmall,aes(log(carat),log(price))) +g
eom_point() +geom_abline(intercept=theat[1],slopt
=theta[2],size=2,color=I("red"))
Error in data.frame(intercept = intercept, slope
= slope) :
  object 'theat' not found
> ggplot(diamSmall,aes(log(carat),log(price))) +g
eom_point() +geom_abline(intercept=theta[1],slope
=theta[2],size=2,color=I("red"))
>
```

40. Let's now do the last regression example, which we'll call M4. And M4 will build a regression for the log of the price based on the log of the carat. Let's see what we get. In this case, if we want to look at the coefficients, for example, here are the coefficients. Now let's visualize it and see what happens in this case. So first, we just plot the scatter plot of log price versus log carat in the training set. Now we see a very nice linear linear relationship. So by taking the log of the carat and the log of the price, going to the log scale, or log log scale, both x and y axis, the previously nonlinear relationship now looks much better. Now it looks nice linear relationship and maybe that's the way to go. So rather we transform to a new space, also the response variable, and the measurement variable, and then in that space we have a much nicer looking linear relationship, and that's maybe what we should model using the linear regression. Let's overlay the linear regression line on top of the scatter plot. This is the regression line and we see a very nice fit to the points. And now if we want to explore the model, for example, maybe we want to look at the r.squared coefficient. Or we want to look at the residuals, the residual squared. One thing to notice is that since we transformed the response variable, now the RSS or the residuals, they're not immediately relatable to the previous residuals, because the previous residuals were on a different scale. So we can now try to add more variables in the log price model and see how the RSS and the r.squared changes. But it's going to be a problem to relate these values that we see right here to the values of the r.squared and the residuals that we saw in previous models M1, M2, M3. So we're going to need to compare them using a different methodology, for example, looking at the residuals on the non-transform scale and seeing how the present model M4 performs in that respect.

Defining Linear Models

lm:

- Additive terms are added with plus signs
- Includes the constant term
- Remove the constant term by adding +0
- Use ':' to encode interaction by two terms
- Use '*' for all possible products between groups
- Use '^' for higher powers
- I() to interpret symbols literally
- Drop variables with '-'

41. Let's see a few more options of defining formulas in lm function in R. So we already saw that we can just add additive terms with a plus sign, and that includes by default the constant term or the intercept, which we get if we identify one of the x dimensions automatically with 1. We can remove the constant term by adding +0 to the formula in lm. If for some reason we do not want to have an intercept, we can use a column to encode interactions between two terms. We can use a star or multiplication for all possible products between two groups. We can use the caret sign for higher powers. And we can use I's and parentheses to interpret symbols literally. That is sometimes needed to make sure that R parses the formula correctly. If we want to drop some variables, for example, variables created automatically based on the colon or the asterisk, and we want to remove these terms from the formula, we can dropped them using a minus sign. And we'll see some examples in the next two slides of formulas and the models that they correspond to.

Model Formulas

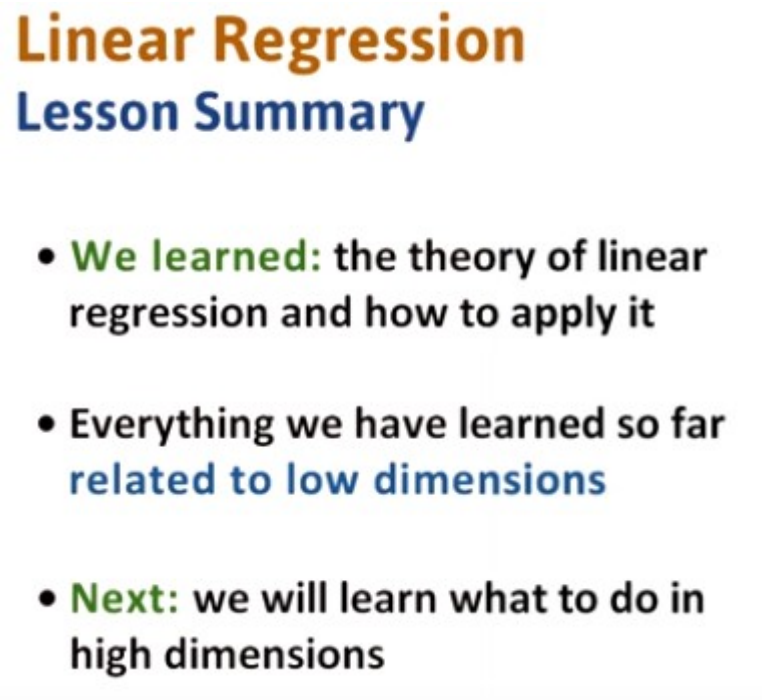
formula	model
$y \sim x$	$y = \theta_1 + \theta_2 x$
$y \sim x + z$	$y = \theta_1 + \theta_2 x + \theta_3 z$
$y \sim x * z$	$y = \theta_1 + \theta_2 x + \theta_3 z + \theta_4 xz$
$y \sim (x + z + w)\hat{}2$	$y = \theta_1 + \theta_2 x + \theta_3 z + \theta_4 w + \theta_5 xz + \theta_6 xw + \theta_7 zw$
$y \sim (x + z + w)\hat{}2 - zw$	$y = \theta_1 + \theta_2 x + \theta_3 z + \theta_4 w + \theta_5 xz + \theta_6 xw$

So here are some examples. So y tilde x, the model is y equals theta 1 plus theta 2 x. If we add to x also z, we have also an addition of theta 3 times z. If we have a multiplication x times z, we add both x and z and x times z automatically to the regression model. And that's what I meant when I said all possible interactions. We can have a power using the caret symbol power of 2 applied to either a single variable, or in this case, applied to a group of variables. In this case, applied to x + z + w, this will expand to a formula of having x and z and w as well as xz, xw, and zw. If we want to drop, for example, the zw term, we can have -zw at the end of the formula, that will drop that term specifically. If we want to remove the intercept term, we can just add a 0. The colon sign includes the interaction between x and z, but does not include x and z as well separately, as will the multiplication or asterisk sign. And so if we do want to have to x and the z and the xz, we can do x + z + x : z or we can just do x times z.

Model Formulas

formula	model
$y \sim x + 0$	$y = \theta_1 x$
$y \sim x : z$	$y = \theta_1 + \theta_2 xz$
$y \sim x + z + x : z$	$y = \theta_1 + \theta_2 x + \theta_3 z + \theta_4 xz$
$y \sim I(x + z)$	$y = \theta_1 + \theta_2 (x + z)$
$\log(y) \sim \log(x)$	$\log(y) = \theta_1 + \theta_2 \log(x)$

If we want to model y as a linear regression based on the addition of x + z as one single measurement, we need to add the I sign in parenthesis so that R interprets this literally, rather than try to expand it as a formula. The default interpretation of R would be to think that x plus z is the plus of the formula expansion rather than the plus of a transformation of variables to create new variables, which is what we intend in this case. And we can transform either the response or the measurement using different transformations, for example, log, in which case we would get the following regression model.



42. In this lesson, we learn how to create linear regression models, use training data, and actually make predictions with the models. You should now feel comfortable using models in r and have a good understanding of the results when applying them. We will see in the next lesson that what we have learned so far applies to low dimensional cases. In high dimensions, we need to use regularization to make sure that our models pick up the signal rather than the noise.