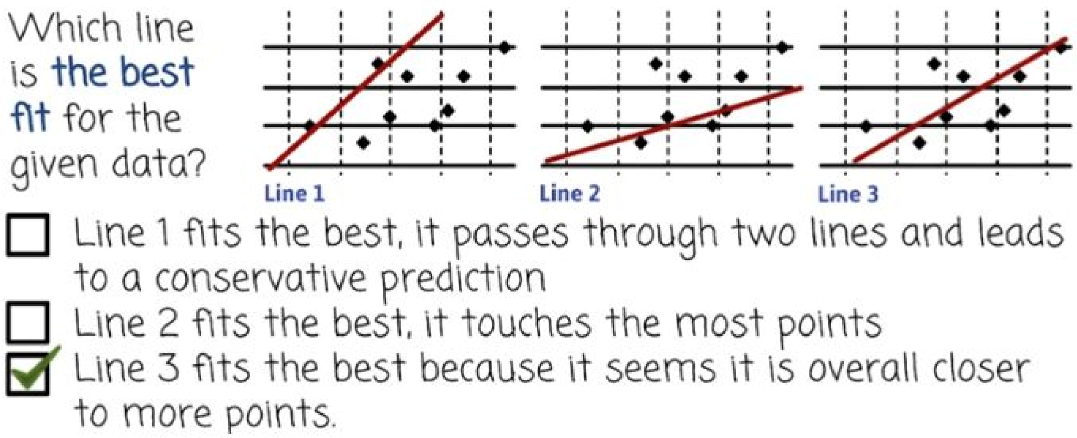
Linear Regression

Notes for CS 6232: Data Analysis and Visualization
Georgia Tech (Dr. Guy Lebanon), Fall 2016
as recorded by Brent Wagenseller

Lesson Preview

- Linear Regression is the task of predicting a real value based on a vector of measurements
- Linear regression
  - Is an important ML algorithm
  - Is used heavily in finance, demand forecasting, and pricing strategies
- We will learn the theory of the most common regression model and how to use it in practice using R

Intuitive Linear Regression Quiz



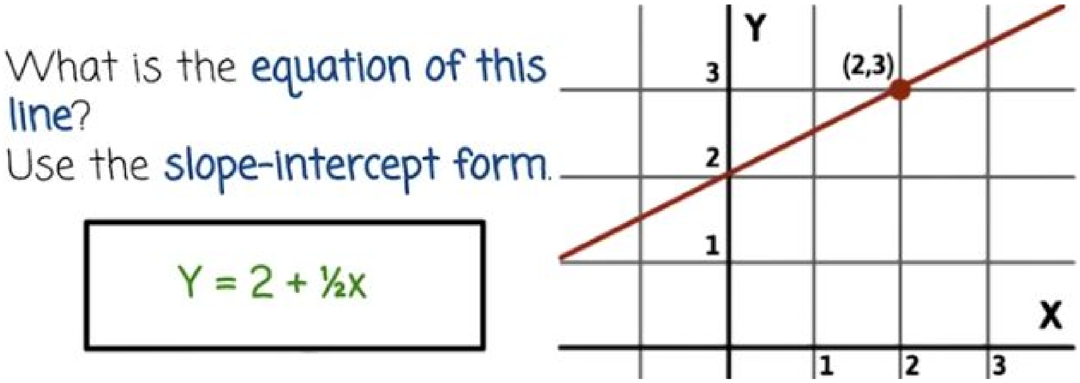
Which line is the best fit for the given data?

□ Line 1 fits the best, it 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.

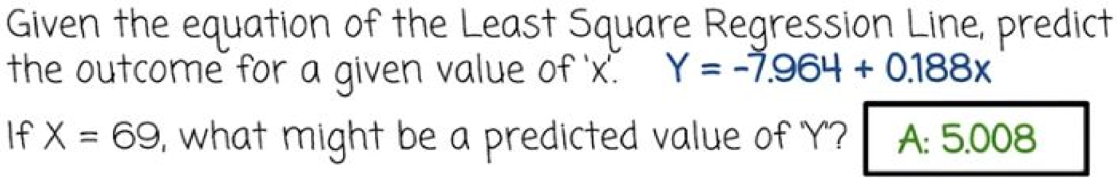
Line Quiz



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

Y = 2 + ½x

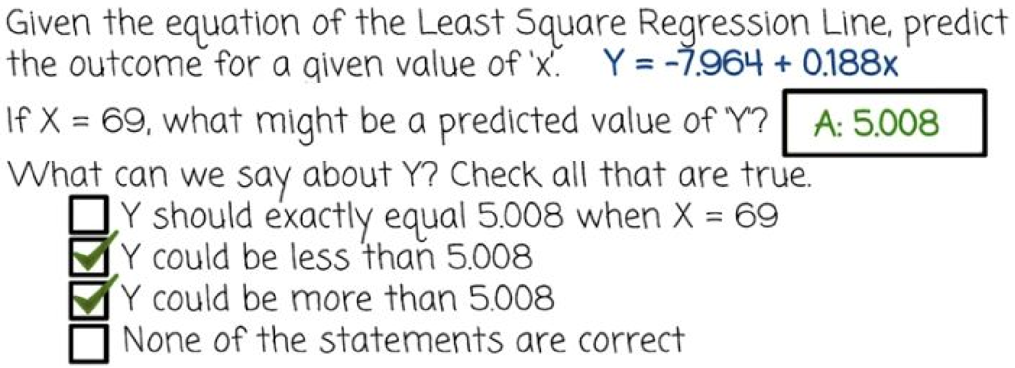
Prediction Quiz



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

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

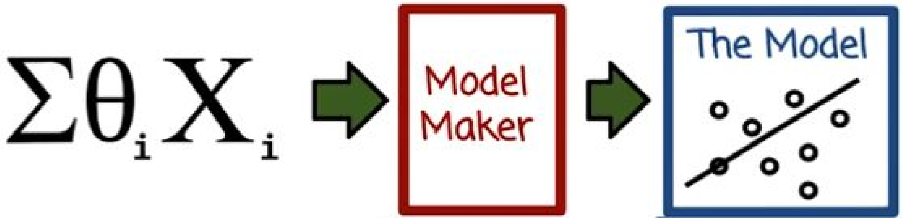
☑ Y could be less than 5.008

☑ Y could be more than 5.008

□ None of the statements are correct

- One thing to note is that linear regression is not precise; it's a prediction (a probabilistic value close to this number but it doesn't have to be exact)
- Linear regression is a probabilistic model which does not make specific deterministic predictions but rather predictions about the probability of Y given X

Linear Regression Model



  - There is a linear combination of variables / features times weights, then summed
    - Similar to logistic regression
    - The theta vector is referred to as 'the model'
    - The 'X' is, again (like logistic regression) a vector of X values
      - Note there is ALSO usually an additional intercept, but not always:

$$\hat{Y} = \theta_1 + \sum_{i=2}^{d} \theta_i X_i = \sum_{i=1}^{d} \theta_i X_i = \theta^\top X$$

      - Note that you will also have to create a 'dummy' X vector fully comprised of 1's (like logistic regression); this is why the equations above are equivalent as its masking this intercept as such
      - The final one – with the transpose – is because we usually assume vectors are column vectors (so the transpose here is just saying make sure its one row, not one column)
    - The 'Y' is a scalar
  - The result is the prediction
- The linear regression model assumes that Y equals $\Theta^T X + \varepsilon$,
  - Also known as:

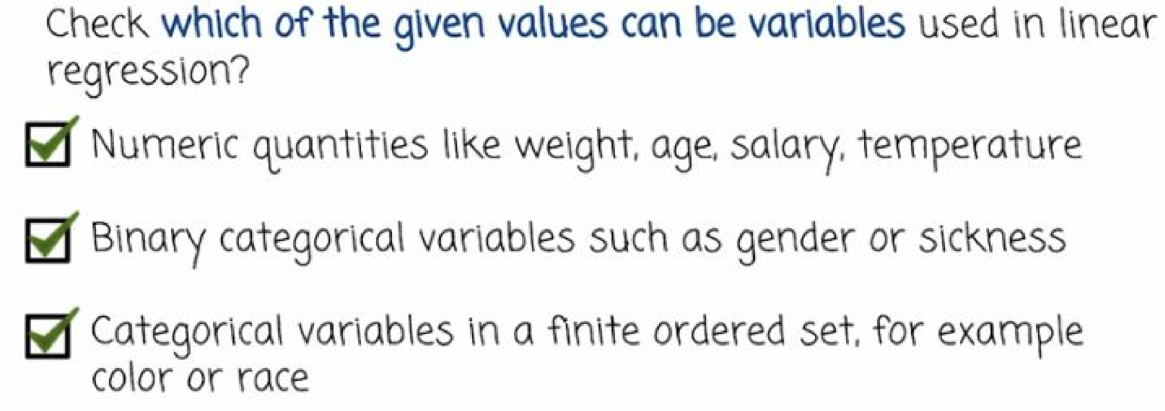
$$Y = \theta^\top X + \epsilon, \qquad \epsilon \sim N(0, \sigma^2)$$

    - ε is a Gaussian random variable with mean zero and variance σ^2 (representing noise)
      - This is an assumption that linear regression makes, but it may not represent reality
  - Equivalently we can say:

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

- The distribution of Y conditioned on X is Gaussian, with mean (or expection) Θ^{T} and with variance σ^2
- In other other words, we can say that given X, Y is normally distributed with a mean that is linearly increasing in X and has constant variance
- In linear (and other) regression models, 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) (that is to say, the conditional distribution p(Y given X)
  - Linear regression makes an assumption on the conditional distribution
    - Its agnostic and does not make an assumption about the distribution p(X), which can be arbitrary; we are specifically not trying to model p(X)
      - As a result, when we obtain the linear regression model we can effectively predict the value of Y from vector X, but we CANNOT predict the distribution of X

Variable Quantiles Quiz



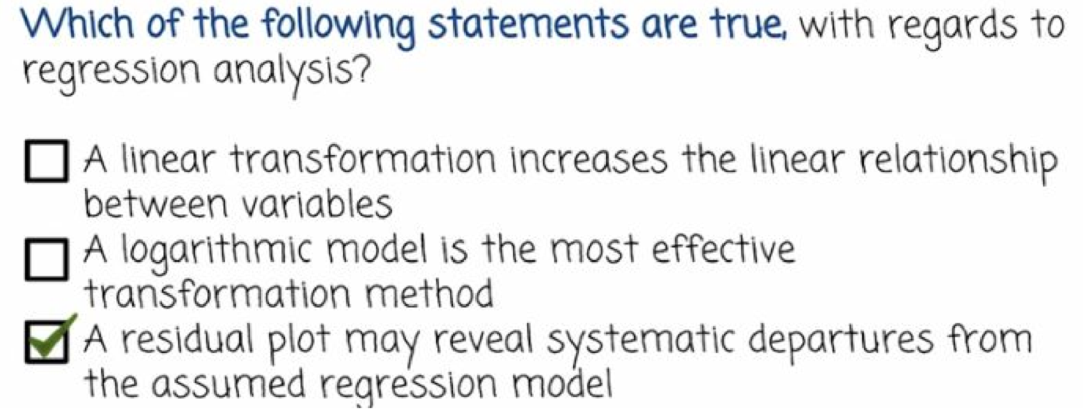
Check **which of the given values can be variables** used in linear regression?

☑ 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

Linear Transformation Quiz



Which of the following statements are true, with regards to regression analysis?

□ A 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

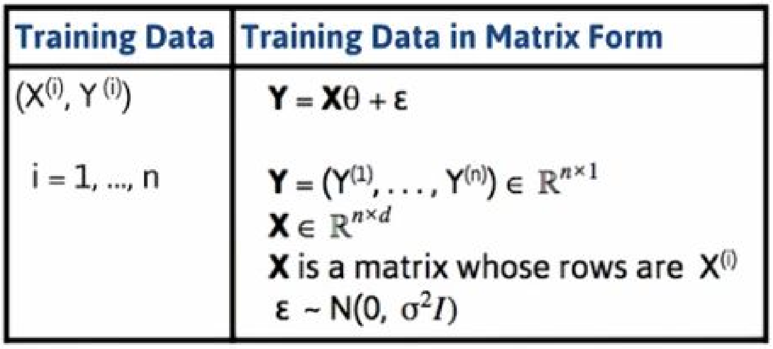
- A logarithmic model may be better or worse…it depends on the model

Training Data

- This is a collection of X pairs and Y labels
  - Same as logarithmic regression

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

  - The pairs are sampled over a joint distribution of X and Y
  - P(Y|X) is the linear regression model
    - This means its expected to have a normal distribution (or mean) of Θ*X and a variance of σ^2
    - This is an assumption
      - In most cases it is not true
        - That said, we can still use linear regression
    - If it IS a Gaussian, we have theoretical guarantees that the training process will produce Θ that will get closer and closer to the Θ that was used to generate the data
  - P(X) is an arbitrary model
- Two ways of getting training data
  - Observational Data
  - Experimental Data
- **Observational Data** involves observing the pair <X,Y> without any intervention from us; in this case, p(X) refers to nature (some process we are not in control of)
- **Experimental Data** is when we are able to interfere or intervene (set the values of X as we wish), in which case X is sampled from p(X), and we control p(X); this is Experimental Design
  - For example, modeling crops from different regions; different areas of planting them
- Matrix Form



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

  - In matrix form, we can express the linear regression model as Y=XΘ + ε
    - Y is the concatenation of all labels we saw in the training data
    - X will be a matrix where the rows are different instances
    - ε is a Gaussian vector with a multivariate distribution with expectation vector 0 and a covariance matrix σ^2*I (I being the identity matrix)
      - ε is the vector of noise values
      - $\varepsilon = Y^{(i)} - \Theta^T X^{(i)} \sim N(0, \sigma^2)$, i=1,…,n corresponding to the training data and is therefore a multivariate normal vector
- Here is a more compact way to represent the linear regression model applied to the data:

$$\mathbf{Y} | \mathbf{X} \sim N(\mathbf{X}\theta, \sigma^2 I)$$

  - This is a matrix (or sequence of row vectors X) has a multivariate Gaussian distribution with expectation vector X*Θ (with X being a matrix and Θ being a column vector) and a covariance matrix sigma squared (which is a scalar) times the identity matrix

How do we get the vector Θ (Minimizing the Sum of Square Deviations)

- In linear regression we have a concept called the **residual sum of squares** (RSS)
- RSS measures error
- How we typically get Θ is by minimizing RSS

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

  - We are finding the theta vector that minimizes RSS with the lowest value
    - Theta<hat> is the result of the training process, which is the vector that can be used for the prediction
  - RSS expressed in matrix Notation:

$$\mathrm{RSS}(\theta) = \|\mathbf{Y} - \mathbf{X}\theta\|^2$$

    - This is the L2 Norm of the vector which is Y – X* Θ squared
    - The same thing, in scalar format:

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

      - This is the square of the ground truth minus the value that the model predicts
      - This is the sum of the squared residuals (residuals is the distance between the predicted and the true value) and we take the square of that
        - The squared is to make sure the negative and positive do not cancel each other out
        - As a consequence, larger mistakes are penalized more heavily
        - We sum the mistakes and this is the criteria we are going to try and minimize
- Θ can be obtained by minimizing the sum of square deviations
  - In logistic regression we saw that the method of maximum likelihood; we can do something similar for linear regression

$$\hat{\theta} = \arg\min_{\theta} \mathrm{RSS}(\theta) \; = \arg\max_{\theta} \textstyle\sum_i \log p(Y^{(i)} | X^{(i)})$$

    - Recall that maximizing the likelihood is the same as maximizing the log likelihood
    - Recall that linear regression models is a Gaussian with a mean of ΘX
    - If we substitute the density of the Gaussian (which involves an exponent) in ($Y^{(i)}$ *$X^{(i)}$), the log and the exponent cancel each other and we just have the squared deviations, which is exactly the RSS
      - The negative comes down, converting the argmin to an argmax
    - In other words, if we maximize the log likelihood we will get minus of the RSS (which turns out to be argmin RSS)
      - This is important as it picks up some theoreticals around the maximum likelihood
        - Specifically consistency (Theta will converge)
        - Efficiency (the convergence is the best possible rate)
- In practice, either maximize the log likelihood or minimize the RSS (same thing) – how do we do this
  - One condition is that the gradient is 0

$$\nabla \mathrm{RSS}(\theta) = 0$$

    - In the case of logistic regression, we had gradient descent
      - In the case of linear regression, the least squared form of the RSS is simpler than the log likelihood of the logistic regression

        - Since this is the case, we can solve this explicitly and see when the gradient = 0
        - We can write this in matrix form

$$\nabla \mathrm{RSS}(\theta) = 0 \quad \Leftrightarrow \quad \sum_i (Y^{(i)} - \theta^\top X^{(i)}) X_j^{(i)} = 0 \quad \forall j$$

            - Note that this assumes we have taken the residual sum of squares and taken the partial derivative (its been done already and was not discussed)
            - This can also be written as a single vector equation:

$$\mathbf{X}^\top \mathbf{X} \theta = \mathbf{X}^\top \mathbf{Y} \quad \Rightarrow \quad \hat{\theta} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{Y}$$

                - In this form we can isolate theta (which is what we want)
                - (Note – to cancel out $X^T X$ you have to take the inverse, which is $(X^T X)^{-1}$

- <u>The end result is</u>

$$\hat{\theta} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{Y}$$

    - Θ(hat) is the solution, signifying the 'correct' theta
        - Again this is the maximum likelihood estimate or the Residual Sum of Squares minimizer
    - This is an explicit formula that relates the data to the vector theta hat
    - <u>There is no iterative process for this</u>
        - That said, <u>in practice the matrix may be large and the inversion can take a long time</u>
            - <u>To do this, an iterative process is usually used – so iterations are not necessarily needed but that changes at a certain point</u>
                - Just note that the iterations are needed to do the inverse, NOT gradient descent
                - This iterative process will not be discussed further
        - If the dimensions are not high, no iterations needed
    - <u>Linear regression is typically simpler and faster than gradient descent (and thus logarithmic regression)</u>
- Further algebra
    - Relate the predicted values to the data
        - AKA model error:

$$\hat{\mathbf{Y}} = \mathbf{X}\hat{\theta}$$

        - Note that Y(hat) are the predicted Y's
    - We can substitute out Θ(hat) for the expression to find the model and end up with HY:

$$\hat{\mathbf{Y}} = \mathbf{X}\hat{\theta} = \mathbf{X}(\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{Y} = \mathbf{H}\mathbf{Y}$$

        - H is simply a substitution and we call it the 'hat matrix'
        - <u>Having it in the more complex version (as opposed to a simple theta hat) allows greater control to do theoretical work</u>
    - One special case of this theoretical work is when the columns of x are orthogonal
        - This doesn't usually happen

        - If it does, the product is simply the inner product of u_j and Y divided by u_j^2 (the norm of u_j)

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

            - In this case u are the individual vectors of X (which are orthogonal)

Coefficient of Determination

- R^2 is the **coefficient of determination**, which is another way to evaluate the model
  - Both R^2 and RSS(Θ(hat)) measure model quality
- Specifically, is the square of the sample correlation coefficient between values Y$^{(i)}$ and the fitted values Y(hat)$^{(i)}$ = Θ(hat)^{2}X R$^{(i)}$
- Equation

$$R^2 = (\textbf{Sample-Correlation}(Y,\hat{Y}))^2 = \frac{\left(\sum_{i=1}^{n}(Y^{(i)}-\bar{Y})(\hat{Y}^{(i)}-\bar{\hat{Y}})\right)^2}{\sum_{i=1}^{n}(Y^{(i)}-\bar{Y})^2\sum_{i=1}^{n}(\hat{Y}^{(i)}-\bar{\hat{Y}})^2}$$

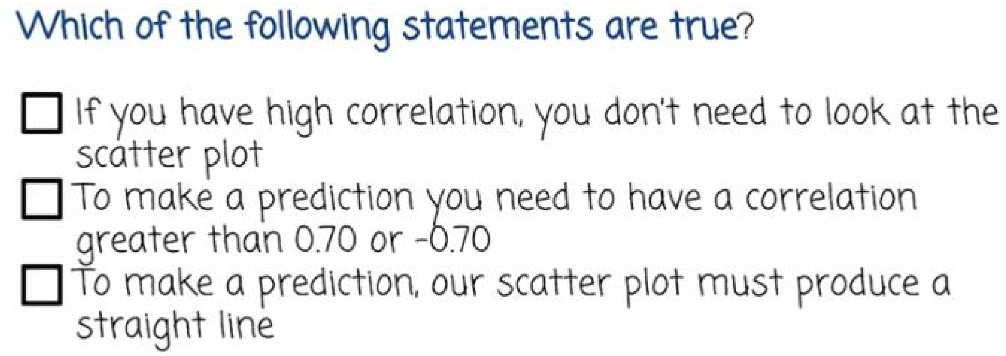
  - In this fraction, the numerator is the covariance and the denominator is the variance of each one of the variables
  - If R^2 is close to 1, it's a very good linear fit
- R^2 and RSS is useful to see if we missed a pattern, which would mean we may want to further transform the features
  - That said they diagnose the fit of the model to the training data, not future test data
    - We also want to look at predictions, residuals, and likelihood values and sample correlation of future values
- R^2 is good because its very interpretable (closer to 0 is bad, closer to 1 is good)
  - Perfect is 1 with no noise (σ -> 0)

Linear Regression in R

- We will use the 'lm' package
  - Stands for 'linear model' formula
  - Usage: lm(linearModelFormula,dataFrame) -> M: object that can be queried
    - The dataFrame holds both the X AND the Y
    - The linear model formula
      - Format:target~xVariable
        - To add variables use a + and then list more variables
        - A multiplication sign can be used to taking all possible products between the different features
          - This is useful to capture all possible interaction between two vectors
          - Every interaction term is converted to a measurement in X
      - R detects which variables are categorical
        - It will transform it to vector X by making it into binary or making it into a binary subsector (a 1 in one component and a 0 in other components a la what we did for movie categories)
        - This means we can keep the data in a nice format – no need to do this manipulation explicitly
    - The model M can be queried in a few ways
      - Using the function predict()
        - predict(M, dataFrame) -> prediction
          - Note the data frame MUST NOT have the Y
        - coef(M)
          - Gets model parameters
          - This is Θ
        - summary()
          - Summary gives a summary of the model and its fit
          - Gives residuals and R^2 values from the training process

- We will use the ‘diamonds’ dataset
    - The target is ‘price’
    - We are going to assume the regression model is:
      Price = theta_1 +theta_2*carat + epsilon
        - Theta_1 is the intercept form
        - This is a 1-dimensional regression
            - There is a single X measurement (in this case, ‘carat’)
        - Epsilon is the noise
- <see the R file for code>

Regression Quiz



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 0.70 or -0.70

□ To make a prediction, our scatter plot must produce a straight line

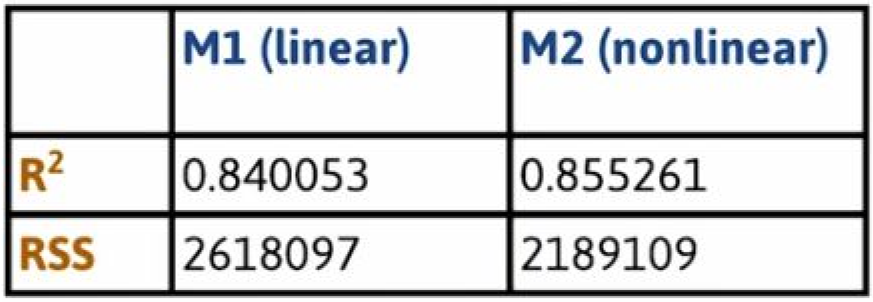
- None are correct
- First one is false as its till useful to view the scatterplot to reveal useful information
- Second is wrong as it can just be a poor model or a lot of noise
    - Just don’t have high confidence
- Third is wrong
    - A nonlinear transformation may be used and the relationship between X and Y which will not lend well to correlation

Adjusting for Non-Linearity

- Its possible that either a nonlinear or a power transformation would help get the data in a more agreeable format for linear regression
- Example (for diamonds)
  Price = theta1 + theta2*carat +theta3*carat^2 + theta4^3 + epsilon
- Note this is not the R equation – see the R lesson model for that
- Its difficult to make predictions that do not have nearby training points
- We use the r.squared coefficients to see that this has a higher value than the last example (which is better)
- We also looked at the average of the squared residuals (mean)
    - mean(residuals(M1)^2)
    - The lower this is, the better the fit

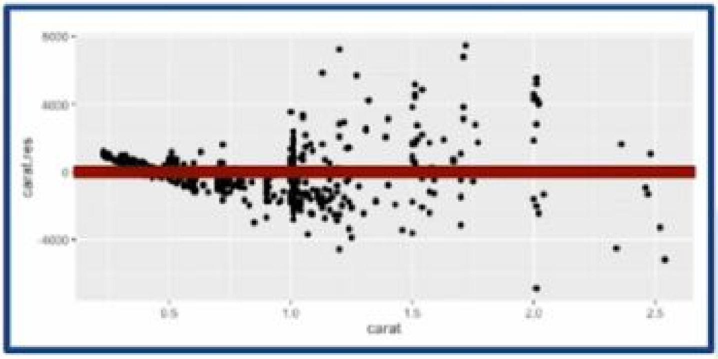
Checking the Fit (of the diamonds example)

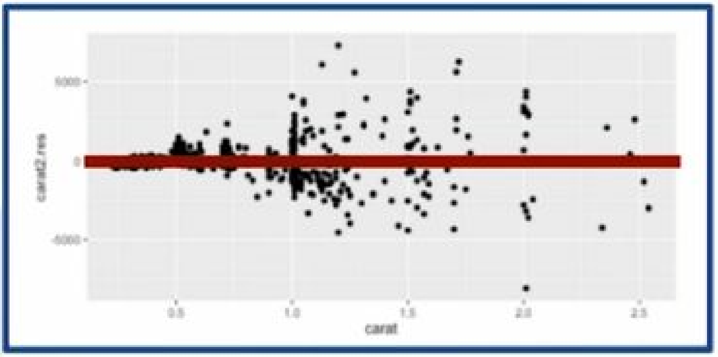
- We model 1 (linear) and model 2(nonlinear) as described in the lesson using R^2 and RSS



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

- To find this its:
  - summary(M2)$r.squared
  - mean(residuals(M2)^2)
- R^2 is higher for M2, which means it’s a better fit
- RSS is lower for M2, meaning it’s a better fit
- M2 is the winner
  - Its nonlinear in carat, but its linear in the transformed space
  - The additional complexity of the nonlinear transformation helped us get a better model
- Below are the residual plots

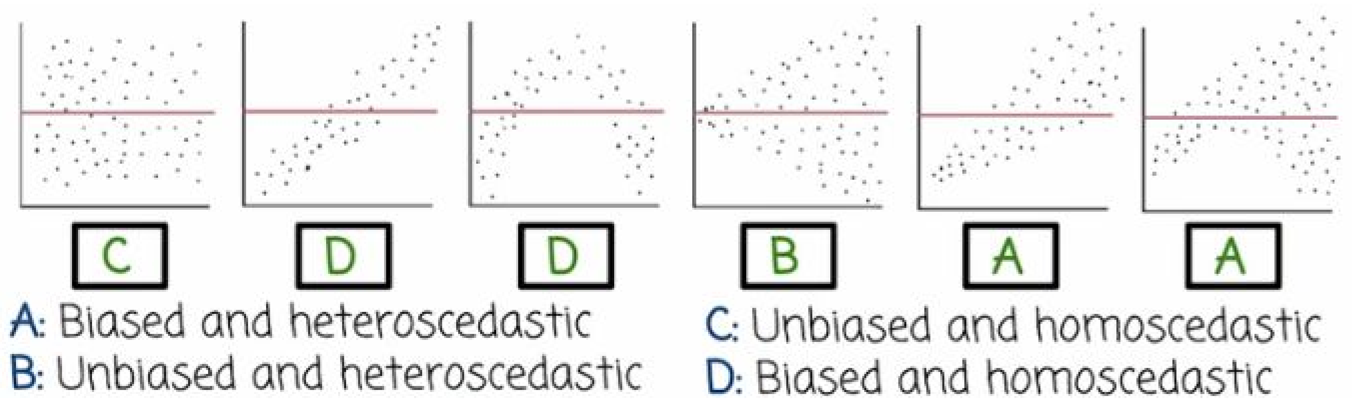




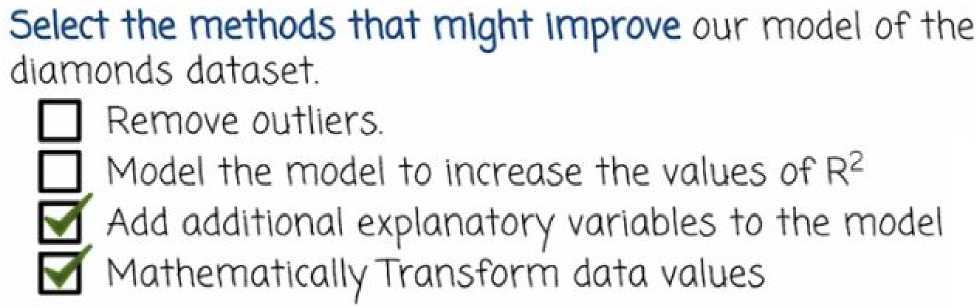
  - Recall that a residual plot is a scatterplot where X is the (single) feature and Y is the difference between the ground truth value of Y and the predicted value
  - M1 is on the right (linear model); M2 is on the left
  - M2 looks much more centered around 0
    - As the carat gets bigger, there is higher spread due to perfect vs imperfect diamonds
    - The residuals increase with the carat

Good Residual Plots Quiz

- **Homoscedasticity** is the assumption that the variance around the regression line is the same for all values of the predictor variable (X)
- **Heteroscedasticity** refers to the circumstance in which the variability of a variable is unequal across the range of values of a second variable that predicts it
- **Unbiased** means the residuals are equally distributed along the regression prediction
- **Biased** means the residuals are no centered around the regression prediction
- Quiz



Improving the model



Select the methods that might improve our model of the diamonds dataset.

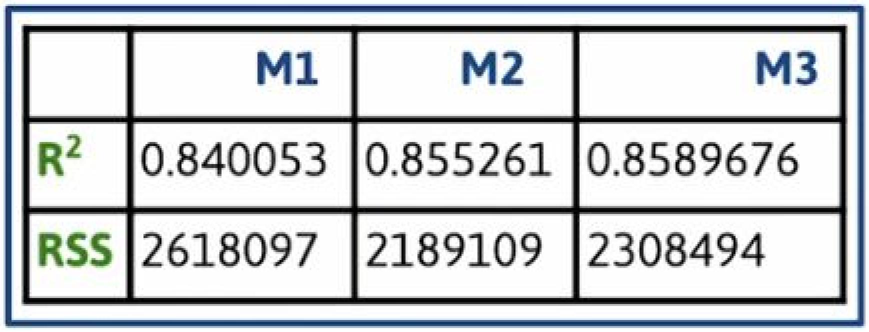
- ☐ Remove outliers.
- ☐ Model the model to increase the values of R^2
- ☑ Add additional explanatory variables to the model
- ☑ Mathematically Transform data values

- Removing outliers can possibly help, but it did not for this specific set

Adding an Explanatory Variable

- This test adds another variable (color)
- ‘Color’ is a categorical variable
  - R automatically detects it and will handle it itself (using what was described previously)
- Formula
  - myModel <- lm(price~carat+color,diamSmall)
- Call this one M3
  - M3, for the instructor, did better on r.squared and the mean of the residuals squared

Comparing The Three Models



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

- Recall
  - M1 = just caret
  - M2 = transformations of caret
  - M3 = caret+color
- It seems M2 had the best RSS but M3 had the best R^2
  - Overall, M2 is probably best

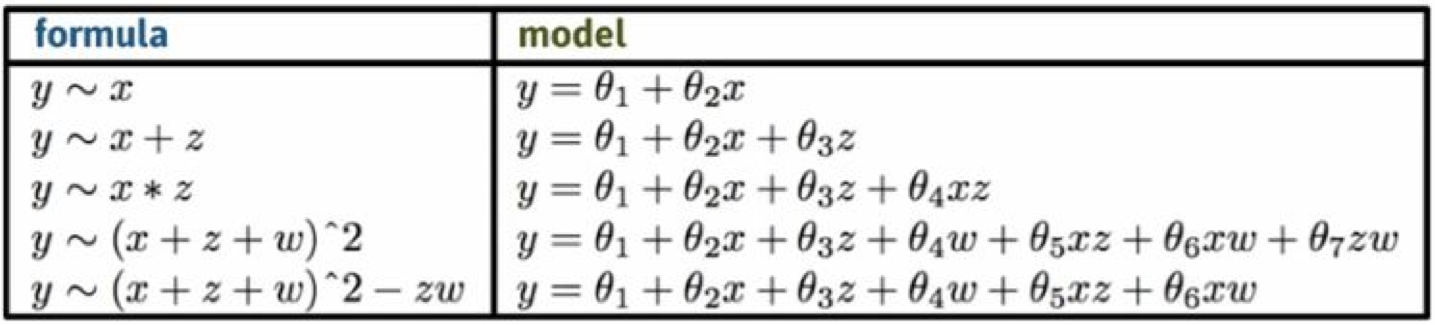
Log Model

- This will predict the log of the price as a linear function of the log of the caret

- Using this method, the previously nonlinear relationship is now remarkably linear
- Finding R and the RSS
    - R is actually higher than the other models
    - RSS cannot be compared, since it’s a log()

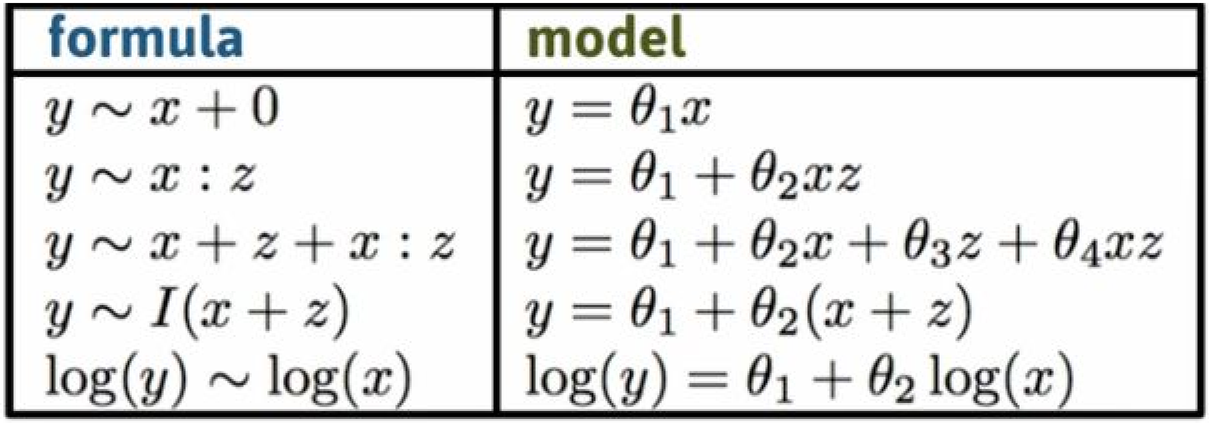
Linear Model Formulas

- lm
    - Additive terms are added with plus signs
        - By default, the constant/intercept is included
        - That said, the constant / intercept can be removed by adding a ‘+0’ at the end
    - We can use ‘:’ to encode interaction by two terms
    - Use “*” for all possible products between two groups
    - Use ‘^’ for higher powers
    - I() to interpret symbols literally
        - Sometimes needed to make sure R parses the formula correctly
    - Drop variables with ‘-‘
- Examples



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

- It seems ^2 does something similar to *?
- The ‘-‘ removes a pairing



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

- The ‘;’ sign includes the interaction, but not the single x or z
- The I() makes it so the + is actually interpreted as a plus

Lesson Summary

- We learned the theory of linear regression and how to apply it
- We learned how to use training data to make predictions