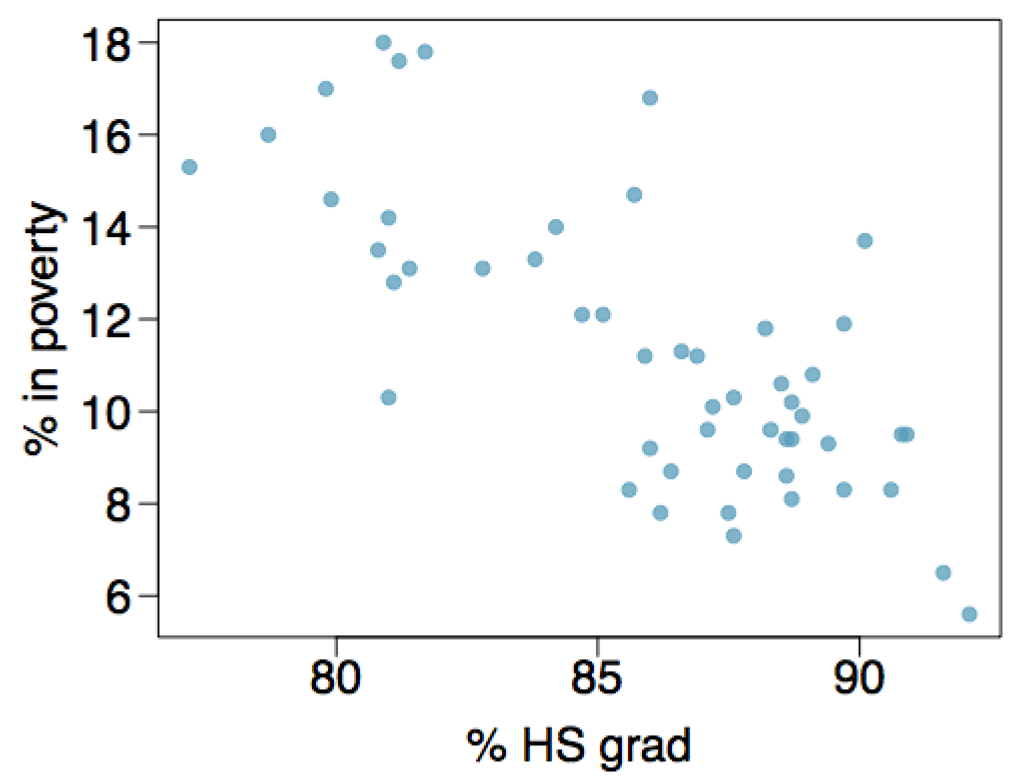
REGRESSION

LINEAR REGRESSION AND PREDICTIVE MODELS

- Most people have done linear regression before – it is making a line of best fit.
  - Result – $y = m*x + b$ line.
  - M = slope, b = y intercept.
- **This process is also a simple predictive model – we provide X and get a prediction for Y.**
- The "regression calculation" uses the training data to "learn" how to generate Y from X.
  - That's the machine learning bit.
- The process of creating a regression is almost the same as other models, we'll do later.

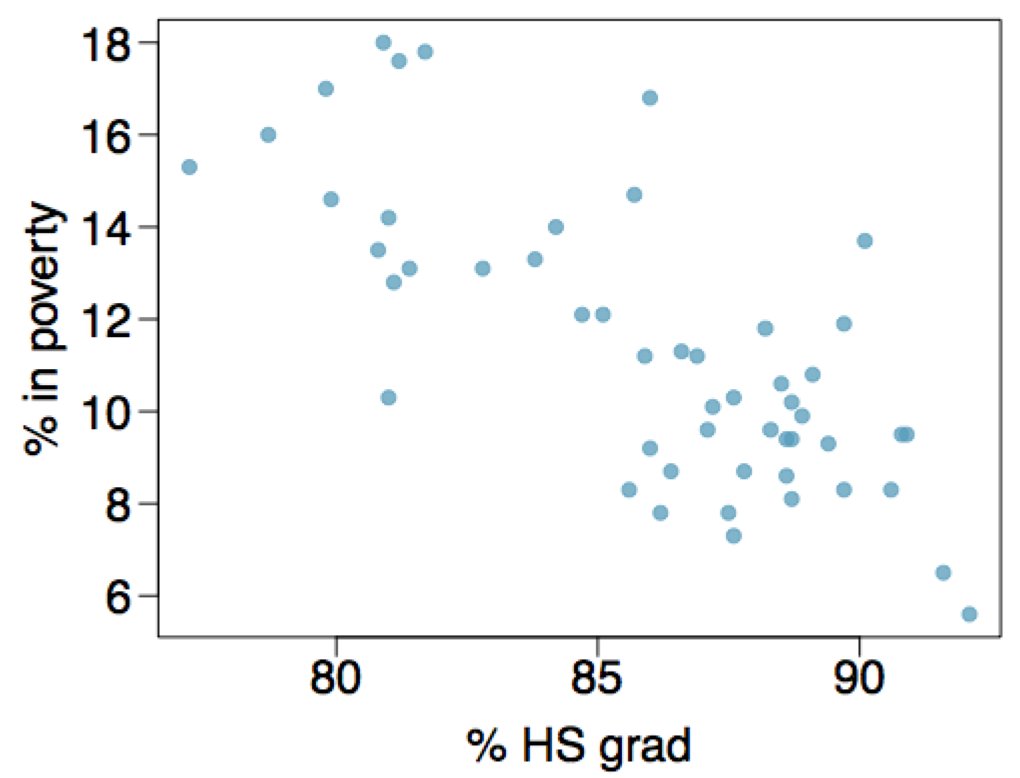
POVERTY VS. HS GRADUATE RATE

The *scatterplot* below shows the relationship between HS graduate rate in all 50 US states and DC and the percent of residents who live below the poverty line
(income below $23,050 for a family of 4 in 2012).



POVERTY VS. HS GRADUATE RATE

The *scatterplot* below shows the relationship between HS graduate rate in all 50 US states and DC and the percent of residents who live below the poverty line
(income below $23,050 for a family of 4 in 2012).



Response variable?

% in poverty

Explanatory variable?

% HS grad

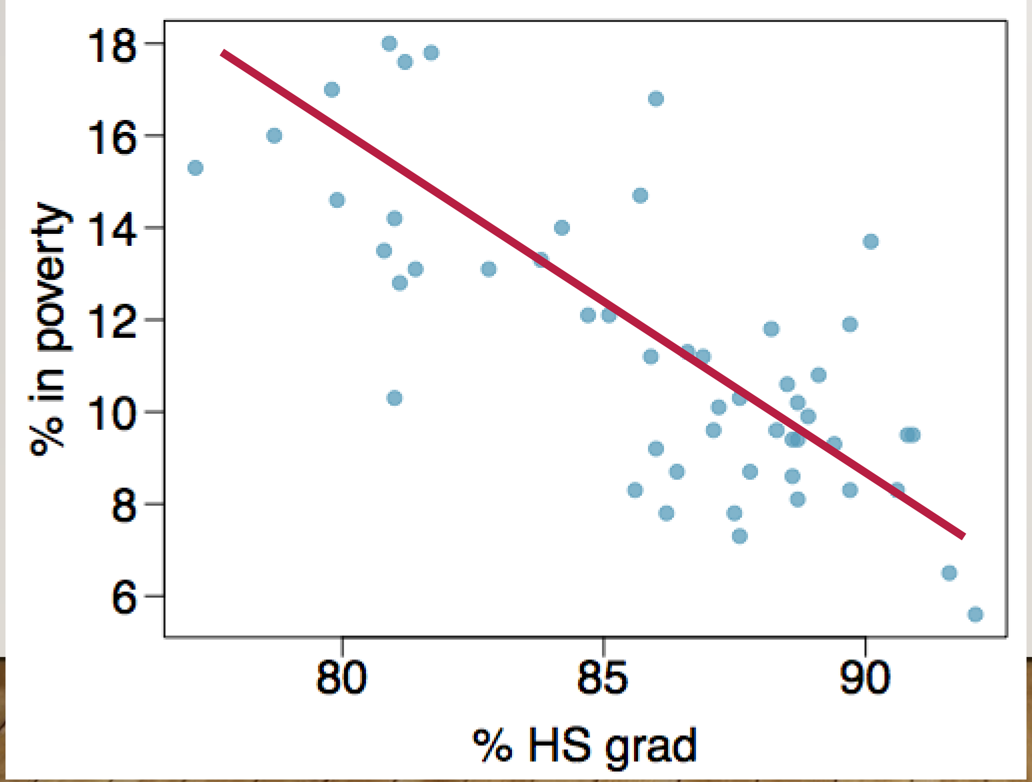
Relationship?

linear, negative,

moderately strong

POVERTY VS. HS GRADUATE RATE

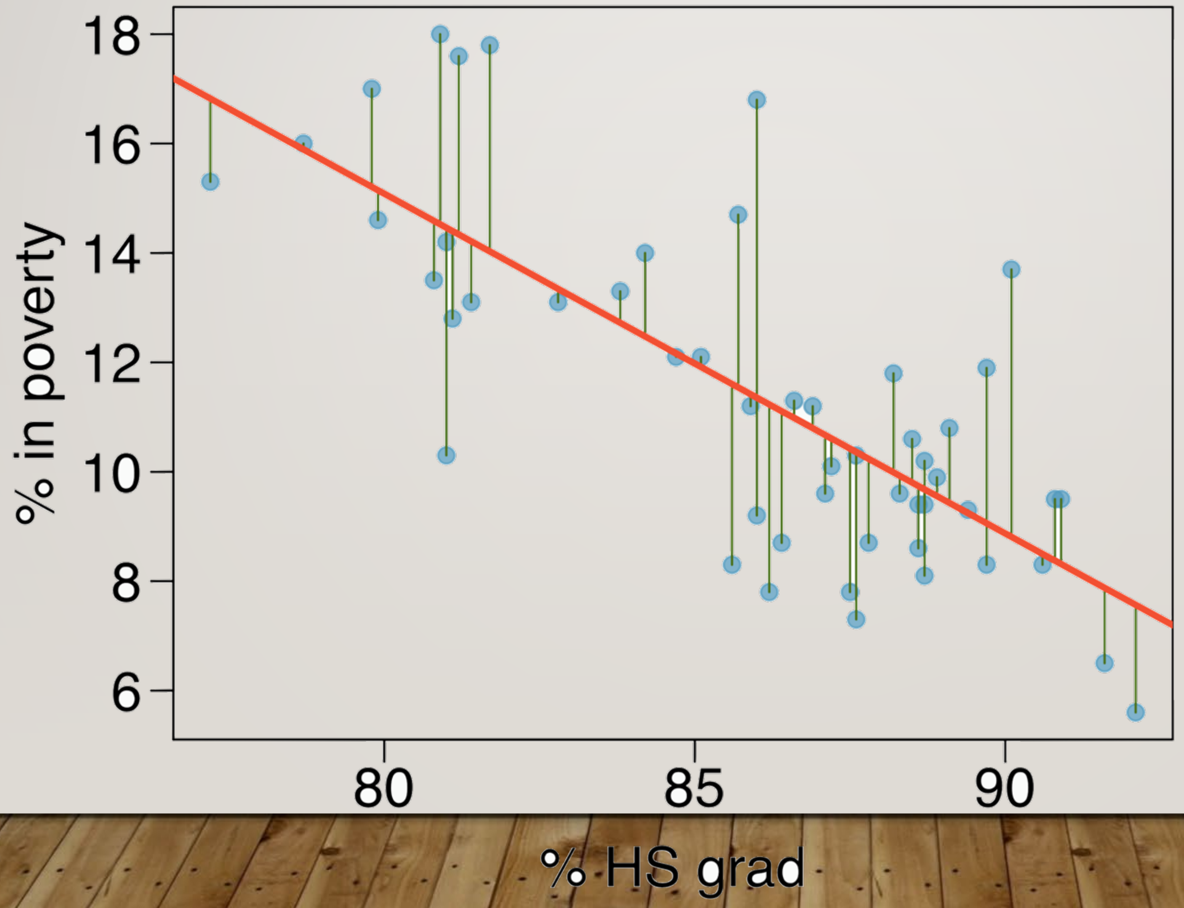
We could draw a line of best fit… but how do we know exactly where it goes?



RESIDUALS

Residuals are the leftovers from the model fit:

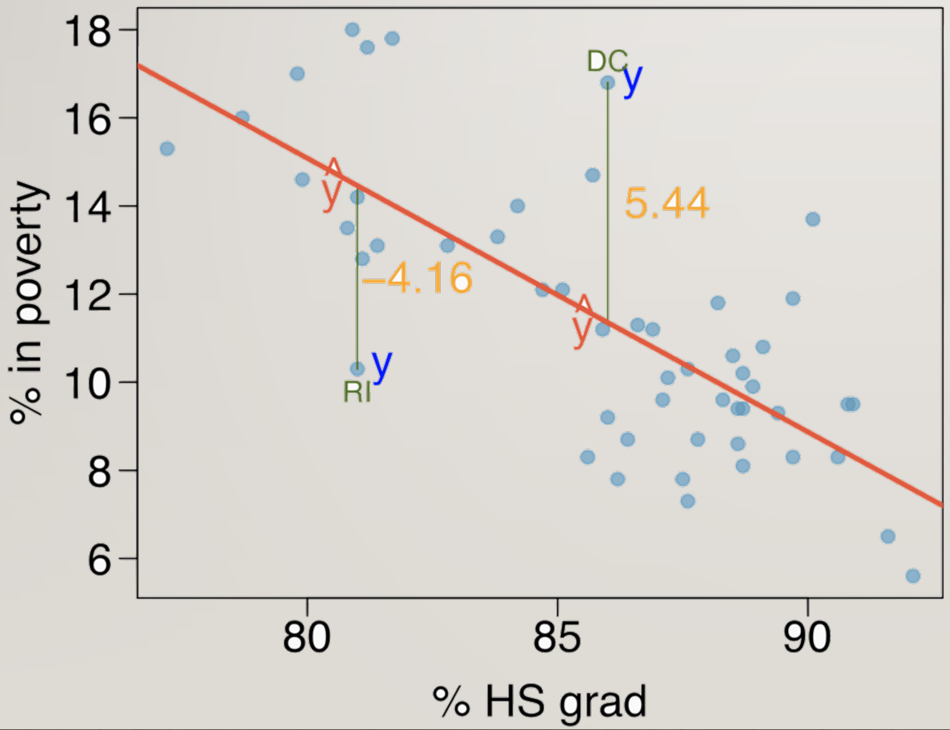
Data = Fit + Residual



RESIDUALS (CONT.)

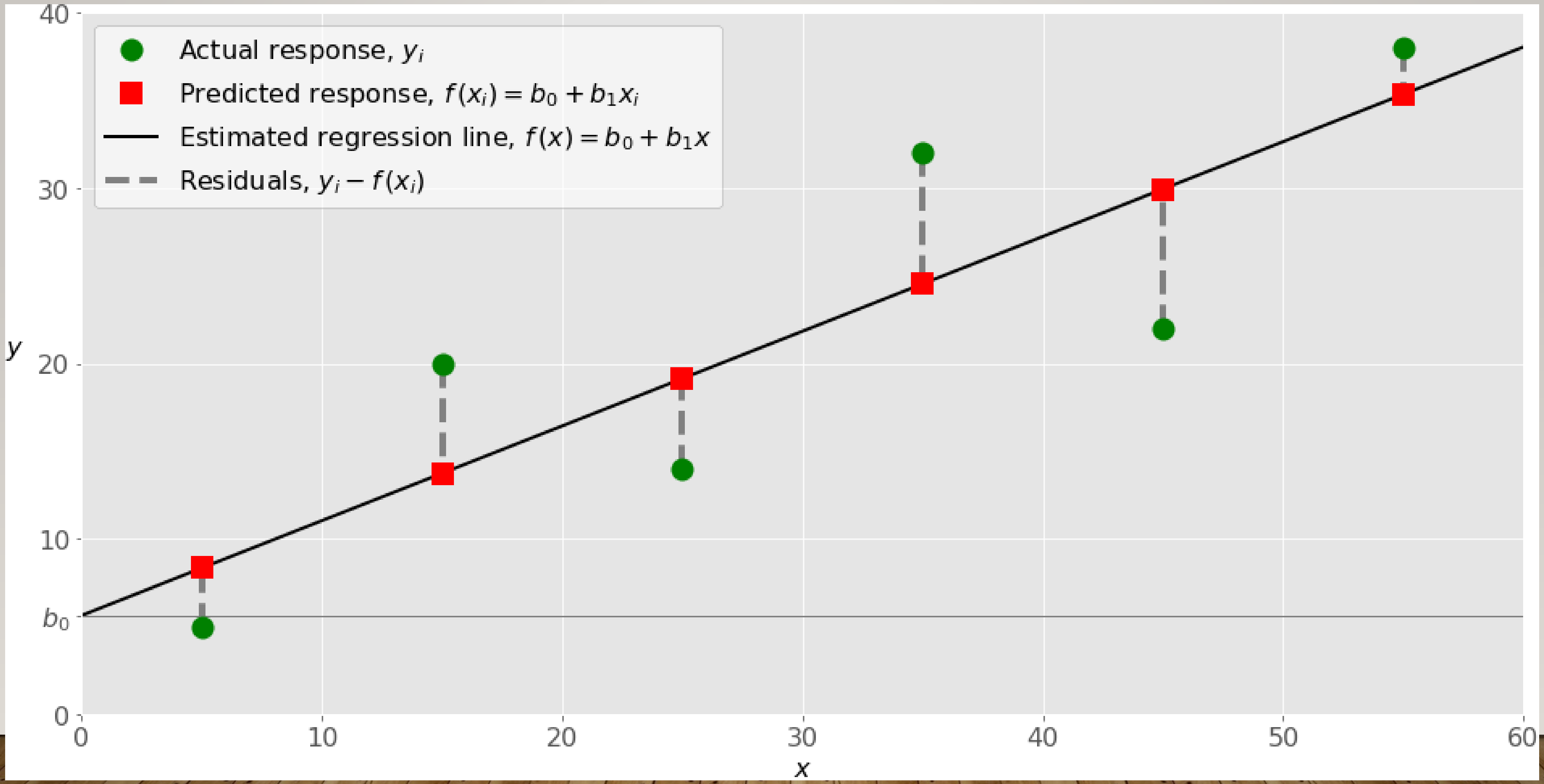
Residual is the difference between the observed (y_i) and predicted $\hat{y}_i$.

$$e_i = y_i - \hat{y}_i$$



% living in poverty in DC is 5.44% more than predicted.

% living in poverty in RI is 4.16% less than predicted.



Actual response, y_i
Predicted response, $f(x_i) = b_0 + b_1 x_i$
Estimated regression line, $f(x) = b_0 + b_1 x$
Residuals, $y_i - f(x_i)$
y
x
b_0
0
10
20
30
40
50
60

LINEAR LEAST SQUARES

- The line can be defined by the intercept and slope.
- We generate a line that minimizes the square of the residuals.
- Why?
    - Small differences matter less than big ones.
    - Squaring deals with negatives.
    - Computationally efficient. (Mattered more in the past)
    - Is (potentially) a good estimator for slope and intercept.

HOW TO FIND THE MINIMUM OF THE RESIDUALS SQUARED?

- This is the "learning" part of machine learning.
- This step is the main thing that differs between other models – the math changes.
- We can use:
  - LeastSquares from thinkplot.
  - StatsModels function.
  - Scipy functions.
  - Probably many other packages.
- The model (best fit line) is defined by the slope and intercept.
  - Add any X value to those two and you can predict Y.
  - The training process finds the "best" calculation to do so.

```
def LeastSquares(xs, ys):
    meanx, varx = MeanVar(xs)
    meany = Mean(ys)

    slope = Cov(xs, ys, meanx, meany) / varx
    inter = meany - slope * meanx

    return inter, slope
```

POVERTY VS. HS GRADUATE RATE

The linear model for predicting poverty from high school graduation rate in the US is

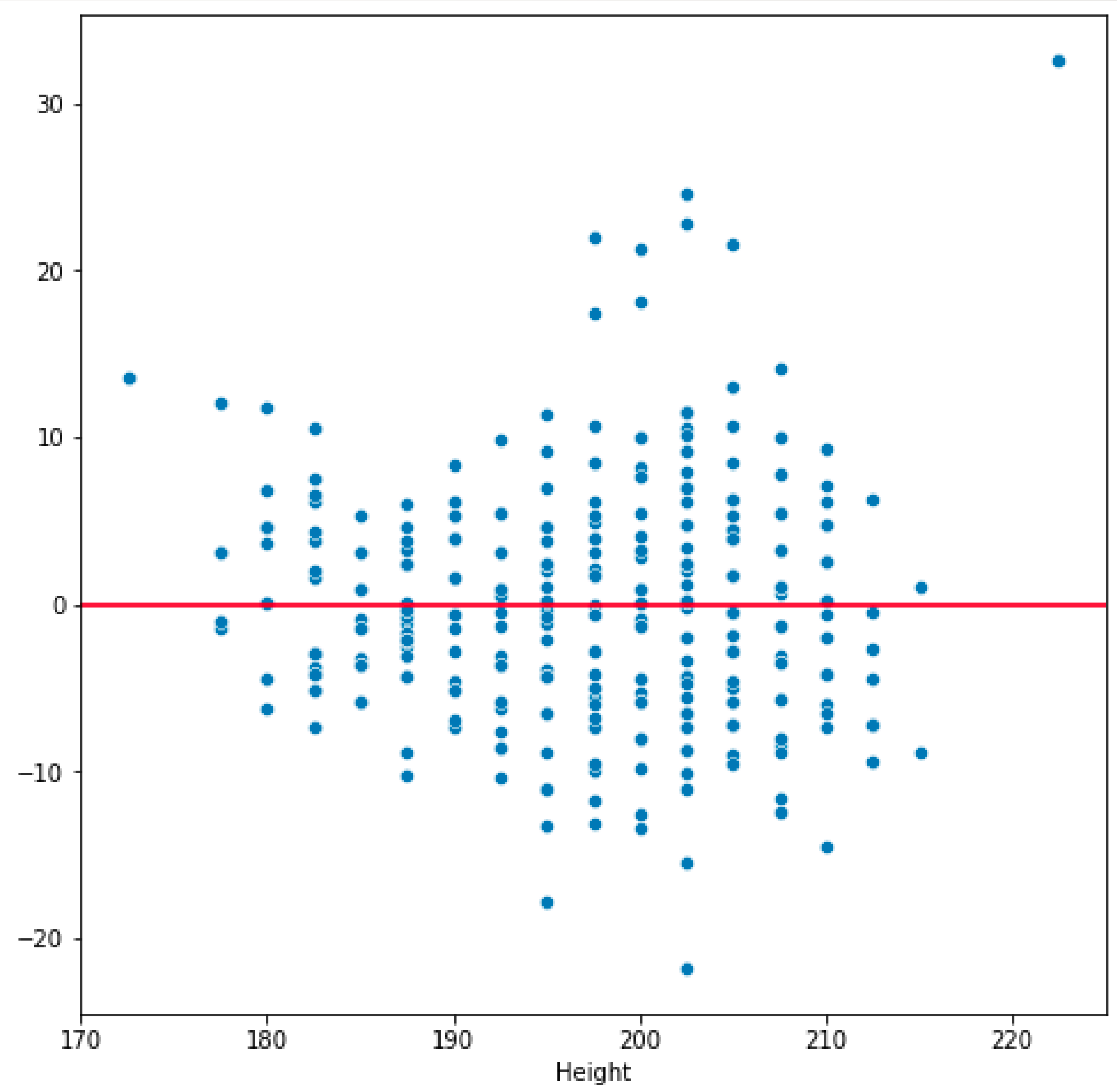
$$\hat{poverty} = 64.78 - 0.62 * HS_{grad}$$

The "hat" is used to signify that this is an estimate.

It is an estimate because this isn't a definitive calculation to calculate the value of poverty – it is a prediction of what we expect the rate of poverty to be, given a value for HSgrad.

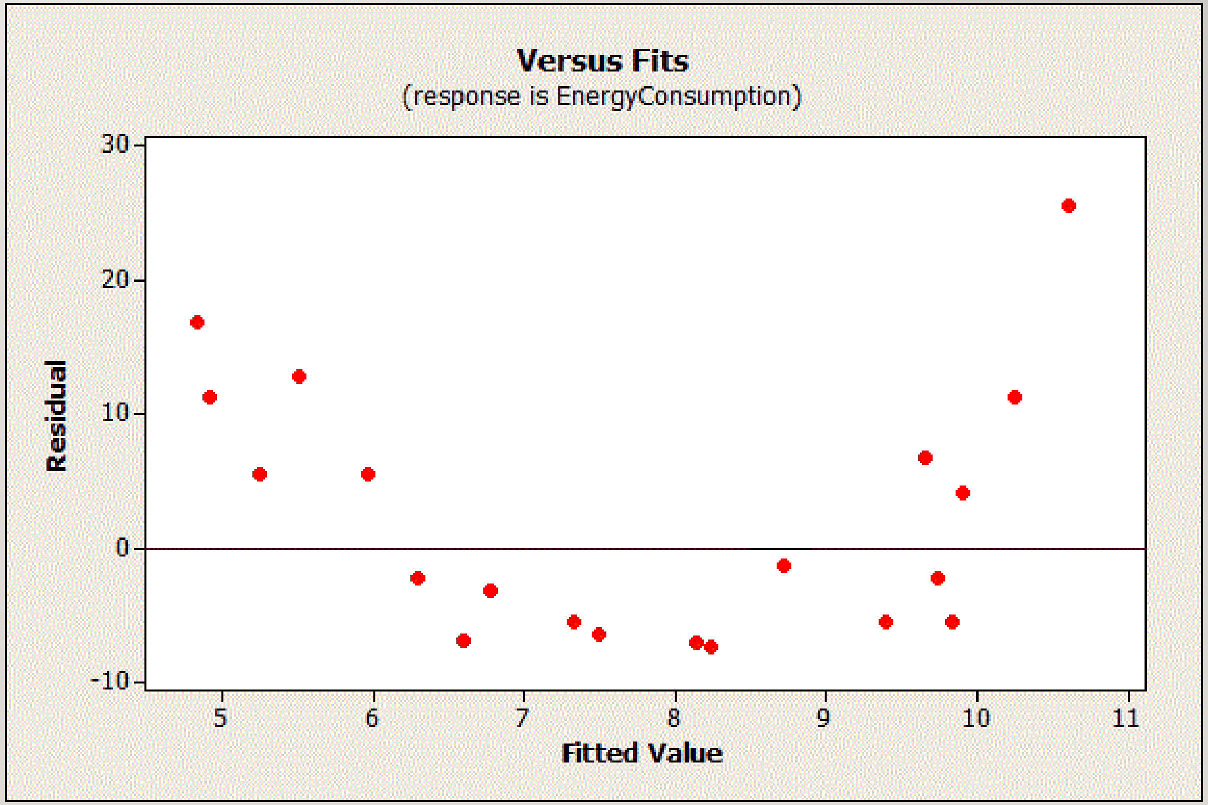
RESIDUAL ANALYSIS

- The generated residuals are also helpful to us in a few ways.
- We can graph the residuals along with X to examine.
- We want this pattern of residuals to not have any patterns in it – to be more or less randomly spread out.
- Why?



WHY RANDOM?

- If there's a pattern in the residuals it tells us that there's some relationship here that isn't captured in our actual model.
  - Middle predictions too high, ends are too low.
  - This pattern should be in the model!
- Residuals should be:
  - Uncorrelated with a variable.
  - Uncorrelated with each other.
- We shouldn't be able to predict residuals.

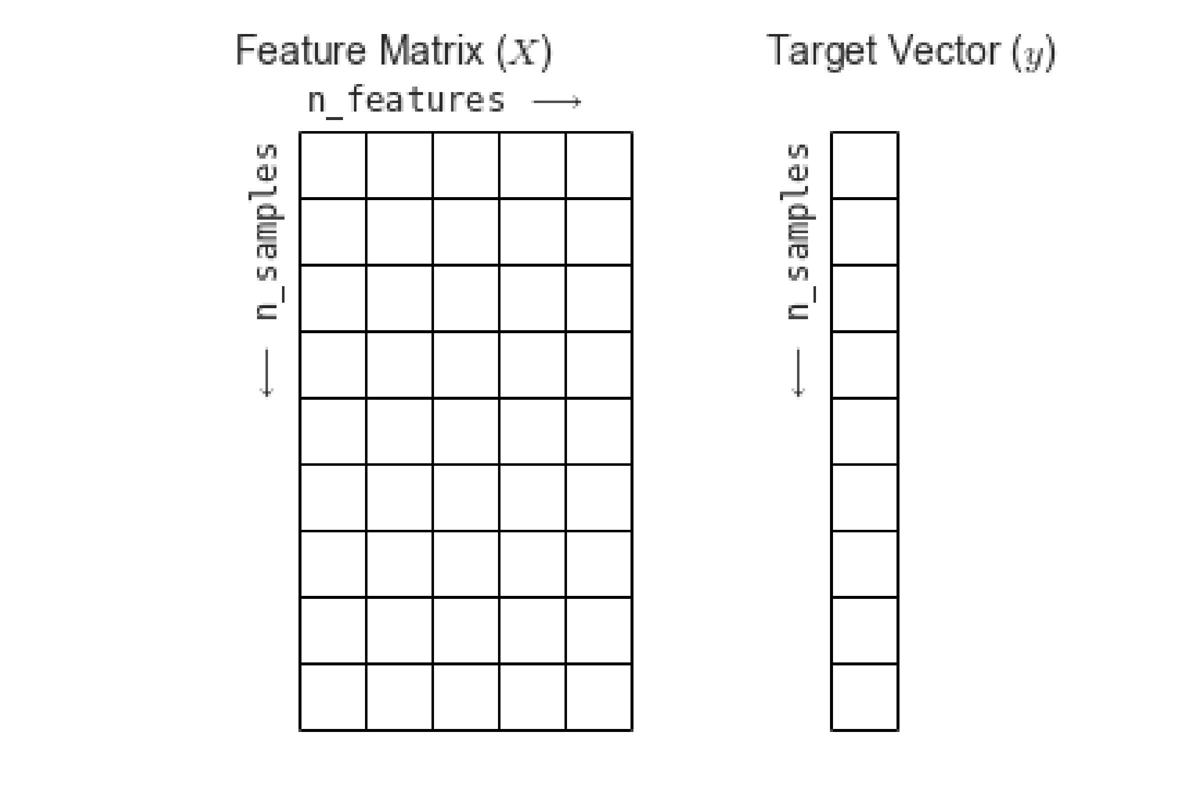


OTHER PACKAGES

- Linear regression is performed by many existing packages, such as StatsModels, Scipy, and Scikitlearn.
- The book uses StatsModels when multiple regression starts.
- Which you use mostly doesn't matter, it is a personal choice.
- We'll use both StatsModels and Scikitlearn:
  - Statsmodels provide more stats data in the output, so we will use that sometimes.
  - The scikitlearn is probably more relevant experience for ML stuff.
- I think going forward I might replace some of the statsmodels examples in future workbooks with sklearn one. The interface is easier, and it is more relevant to ML.

SHAPES AND ARRAYS

- One thing we need to pay attention to a bit more is the data structure and the shape.
- Most things we've used take anything 'iterable' or anything that is list-like.
- Often (but not always) in machine learning we need arrays, usually of a certain shape.
- Some tangible differences are:
    - Use np.array() to create arrays of the data – usually one array for x(s), one for y.
    - Ensure the arrays are "vertical", print it and/or use .shape to look.
    - .reshape(width, height) can reshape the arrays to what we need.



CONCLUSION

- We can train our models to predict Y, given X.
  - In this case, the model is a simple algebra equation.
- This is a simple version of all the more complex ML work to come later.
- The residuals give us information on how good our model is.

- Accuracy and reliability of the predictions…. Next time.