COMS BC1016
Introduction to Computational Thinking and Data Science

Lecture 21: Least Squares, Residuals, and Regression Inference

April 15, 2026

11	4/06	Inference		HW 6 - Confidence Intervals
	4/08		Lab 8 - Normal Distribution and Variability of Sample Means	HW 5 Due 11:59PM
12	4/13	Prediction		HW 7 - Sample Sizes and Confidence Intervals
	4/15		Lab 9 - Regression	HW 6 Due 11:59PM **Final Project Proposals** Due Friday, 4/17
13	4/20	Special Topics: Data Ethics (Computing Fellows Workshop)		HW 8 - Linear Regression
	4/22	Special Topics: Bias in AI (Guest lecture by Murad Megjhani)	Lab: Final Project Work Time	HW 7 Due 11:59PM
14	4/27	Prediction		**Final Project Progress Reports** HW 9 - Regression Inference (Optional, Due 5/4)
	4/29	(Either special topics or prediction)	Lab: Final Project Consultations	HW 8 Due 11:59PM
15	5/04	Wrap up!		HW 9 Due 11:59PM (Optional) **Final Project Report** Due Friday 5/8

Logistics

- **Extra credit opportunity**: Olive's talk tomorrow at 12pm in Milstein 402
  - Worth 5% on the lowest non-dropped homework (comes out to 0.25% of the final grade)
- Submit your Final Project Proposals by Friday on Gradescope as a group
  - **All of your final project components should be submitted as a Python notebook (.ipynb)**
  - **Update**: Project proposal may be submitted late with the late penalty
    - Progress report cannot be accepted late since we need to review them before lab
    - Final Project Report cannot be accepted late due to deadline to submit final grades

Upcoming Labs

- This week’s lab will be the last regular lab
- Starting next week the labs will be final project work times
    - Starting next week you should attend the lab sections that you indicated as attending with your lab partner

Final Project Components

Information can be found on the course website:

https://www.eysalee.com/courses/s26/bc1016_final.html

Project proposal (due Friday, April 17):

- Include required graphs and tables in the “Exploratory Data Analysis”
- Write your null and alternative hypothesis

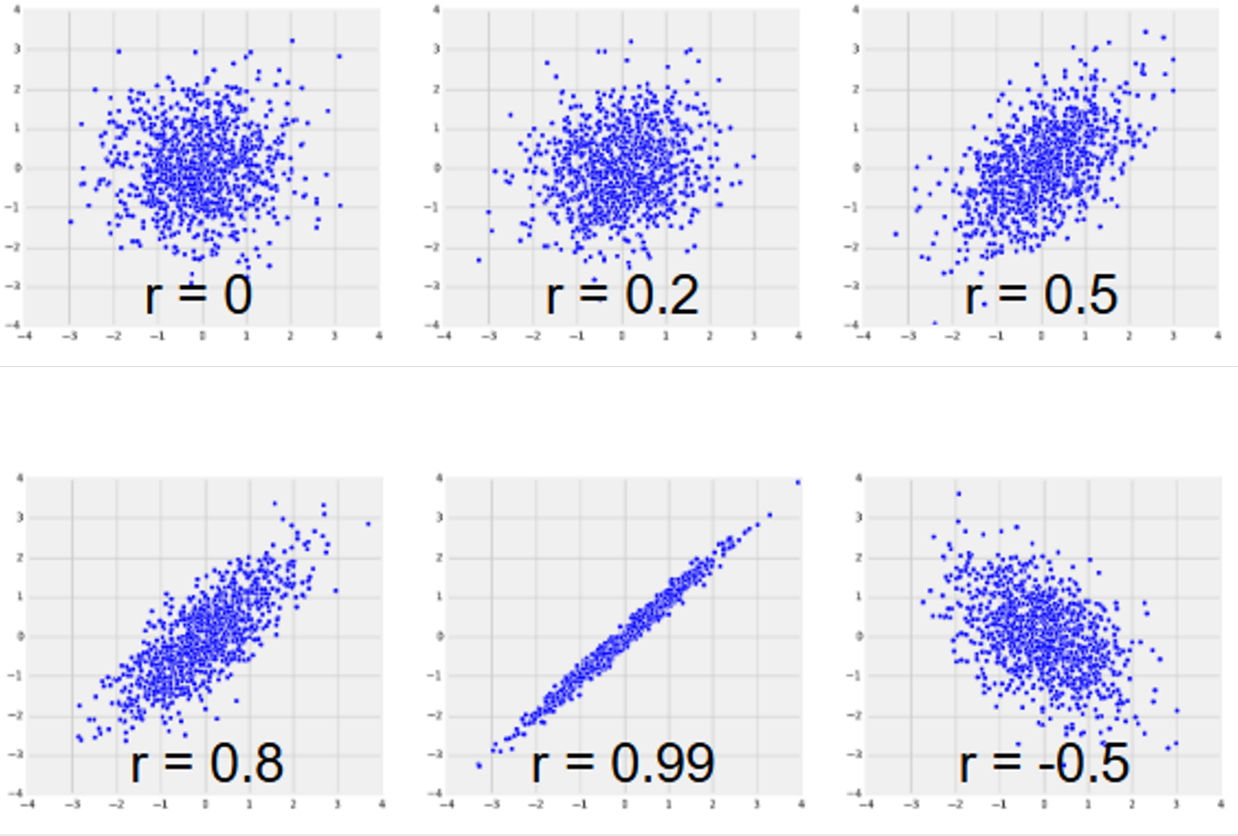
Progress report (due Monday, April 27):

- Complete the exploratory data analysis section (graphs/tables + explanations)
- Complete the hypothesis testing section
- Begin the prediction section
  - Anything you do not complete you should list out what remains and if there are any issues you are running into

Correlation & Regression Line Recap

Correlation Coefficient r

- Measures **linear association**
- Based on standard units
- $-1 \leq r \leq 1$
  - $r = 1$: scatter is perfect straight line sloping up
  - $r = -1$: scatter is perfect straight line sloping down
- $r = 0$: no linear association (*uncorrelated*)



Computing r

The **correlation coefficient** r is the **average product** of x in standard units and y in standard units. To compute:

- First convert our values in x & y to standard units

$$\vec{x}_{\text{su}} = \frac{\vec{x} - x_{\text{avg}}}{\text{SD}_x} \qquad \vec{y}_{\text{su}} = \frac{\vec{y} - y_{\text{avg}}}{\text{SD}_y}$$

- Then compute r as the average product

$$r = \text{avg}\left(\vec{y}_{\text{su}} \times \vec{x}_{\text{su}}\right)$$

Linear Regression Line

The correlation coefficient r can be used to plot the straight line that the points are clustered around:

$$y_{\text{su}} = r \times x_{\text{su}}$$

This is the **linear regression line**

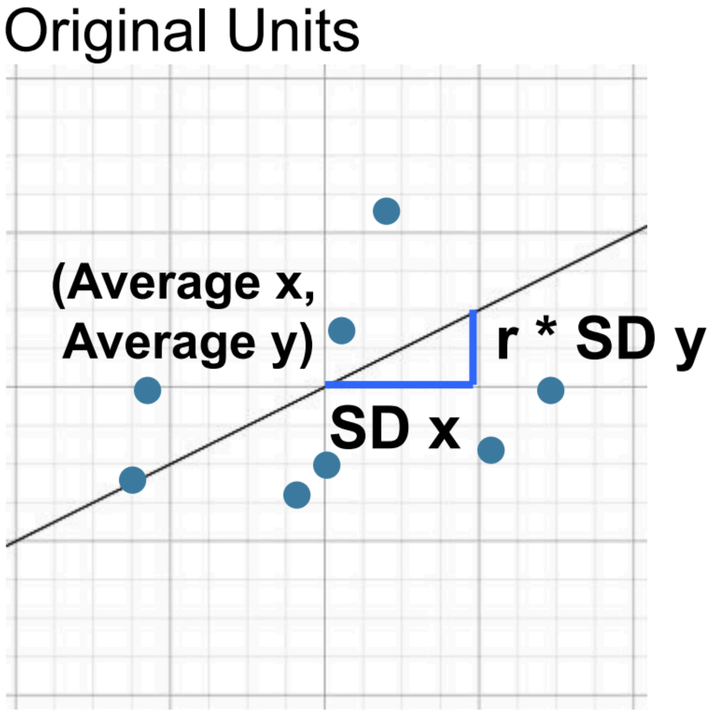
Regression Line: Converting to Original Units

$$y_{su} = r \times x_{su}$$

$$y_{su} = \frac{y - \text{avg}(y)}{\text{SD of } y} \qquad x_{su} = \frac{x - \text{avg}(x)}{\text{SD of } x}$$

$$\frac{\text{estimate of } y - \text{avg}(y)}{\text{SD of } y} = r \times \frac{x - \text{avg}(x)}{\text{SD of } x}$$

Regression Line: Converting to Original Units



$$\text{estimate of } y = \text{slope} \times x + \text{intercept}$$

$$\text{slope} = r \times \frac{\text{SD of } y}{\text{SD of } x}$$

$$\text{intercept} = \text{avg}(y) - \text{slope} \times \text{avg}(x)$$

Prediction with Linear Regression

Goal: Predict y given x

$$z = \frac{v - \mu}{\text{SD}}$$

To find the regression estimate of y:

1. Convert the given x to standard units
2. Multiply by r to get y in standard units
3. Convert y in standard units back to original units of y

$$y_{\text{su}} = r \times x_{\text{su}}$$

Prediction with Linear Regression

Goal: Predict y given x

Examples:

- Predict number of hospital beds using air pollution
- Predict house prices using house size
- Predict number of app users using number of app downloads

Grade Example

A course has a **midterm** (average: 70, standard deviation: 10) and a hard **final exam** (average: 50, standard deviation: 12).

We create a linear regression line to predict what a final exam score would be for a given midterm score.

In this case:

1. What is our y (i.e., what do we want to predict)?

2. What is our x (i.e., we want to predict y given x)?

Grade Example

A course has a **midterm** (average: 70, standard deviation: 10) and a hard **final exam** (average: 50, standard deviation: 12).

We create a linear regression line to predict what a final exam score would be for a given midterm score and correlation coefficient r=0.75.

3. What do you expect the average final exam score to be for students who scored a 90 on the midterm?

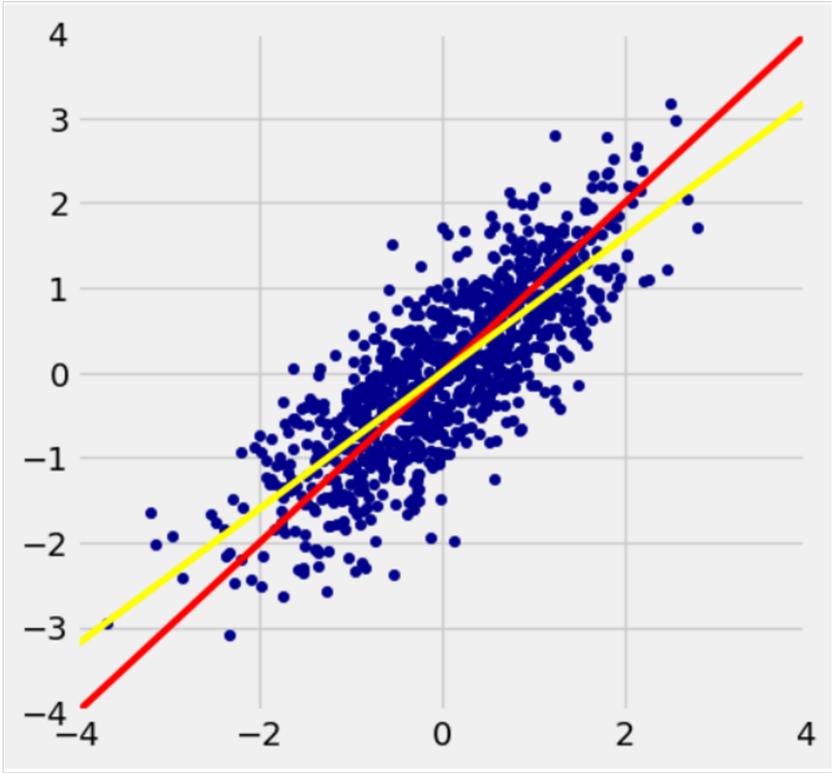
$$z = \frac{v - \mu}{\text{SD}}$$

$$y_{\text{su}} = r \times x_{\text{su}}$$

Least Squares

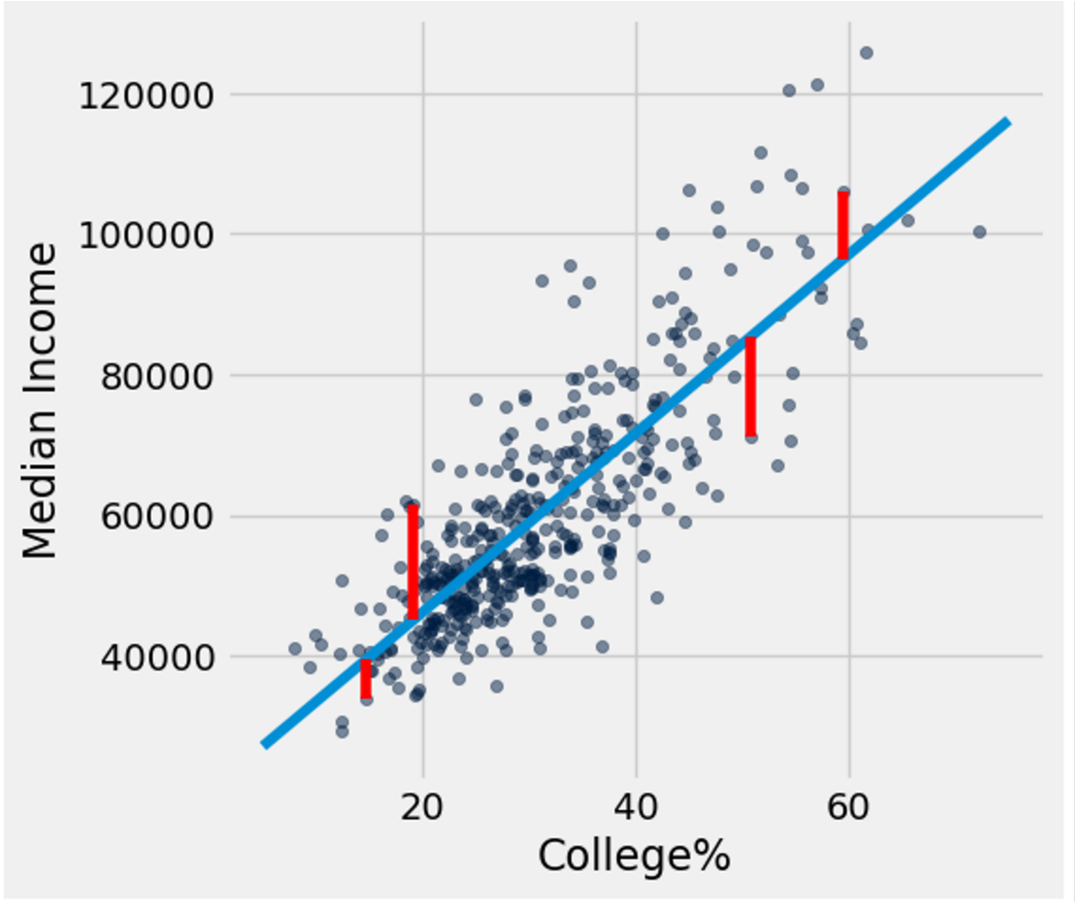
Least Squares

How can we know we've created the best line to fit through our data?

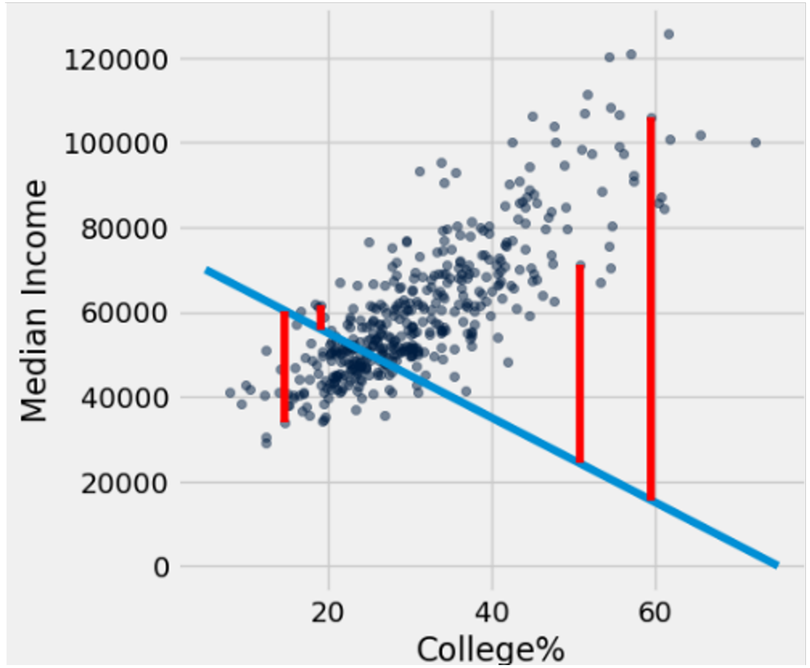


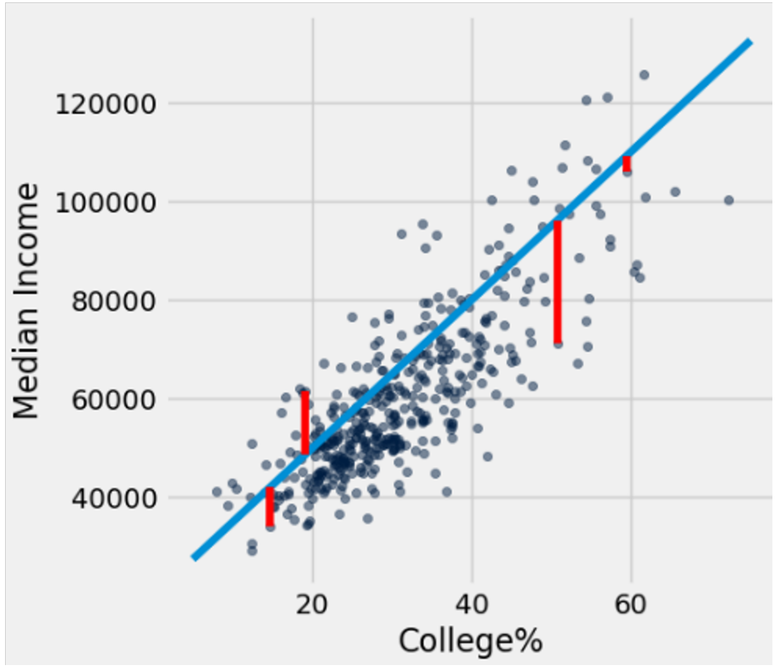
Error in Estimation

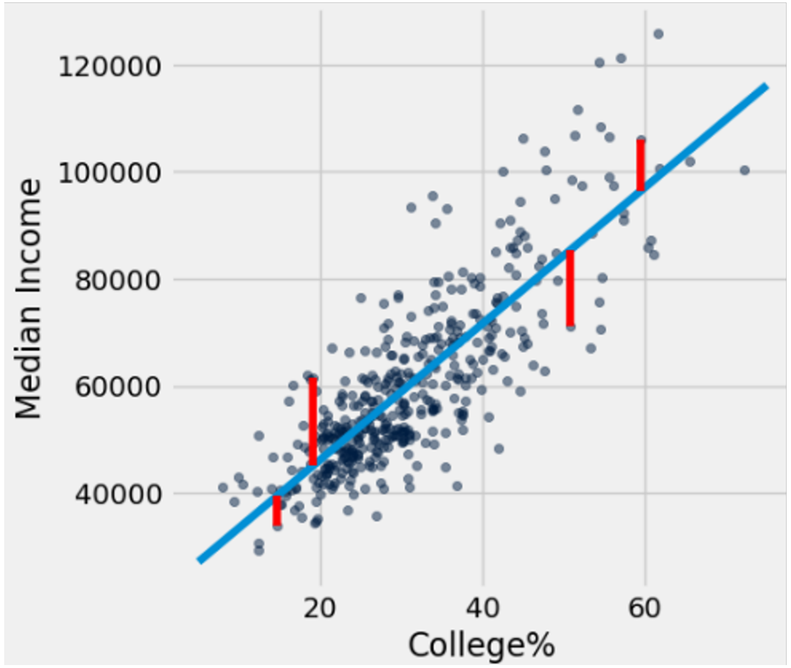
error = actual value - estimate



Different Prediction Lines





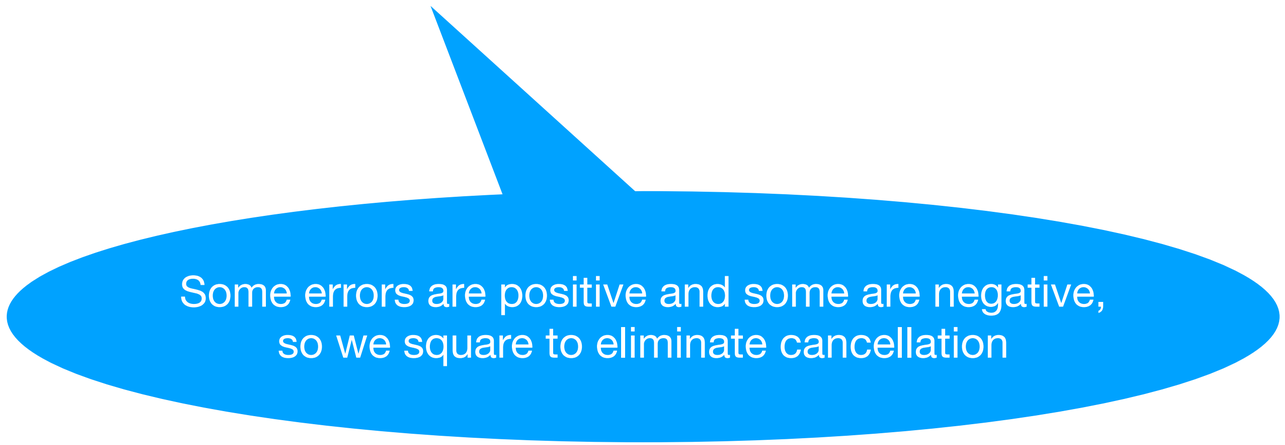


Which one is the best and why?

Root Mean Square Error (RMSE)

Process to calculate the size of the error:

1. Compute the errors between the regression line and actual value and square them



Root Mean Square Error (RMSE)

Process to calculate the size of the error:

1. Compute the errors between the regression line and actual value and square them

2. Compute the mean of the squared errors

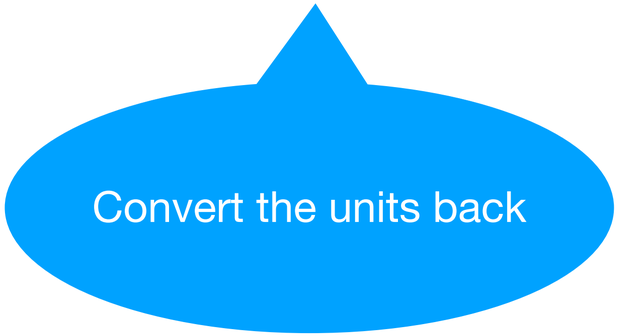
Root Mean Square Error (RMSE)

Process to calculate the size of the error:

1. Compute the errors between the regression line and actual value and square them

2. Compute the mean of the squared errors

3. Compute the square root



Root Mean Square Error (RMSE)

Process to calculate the size of the error:

1. Compute the errors between the regression line and actual value and square them

2. Compute the mean of the squared errors

3. Compute the square root

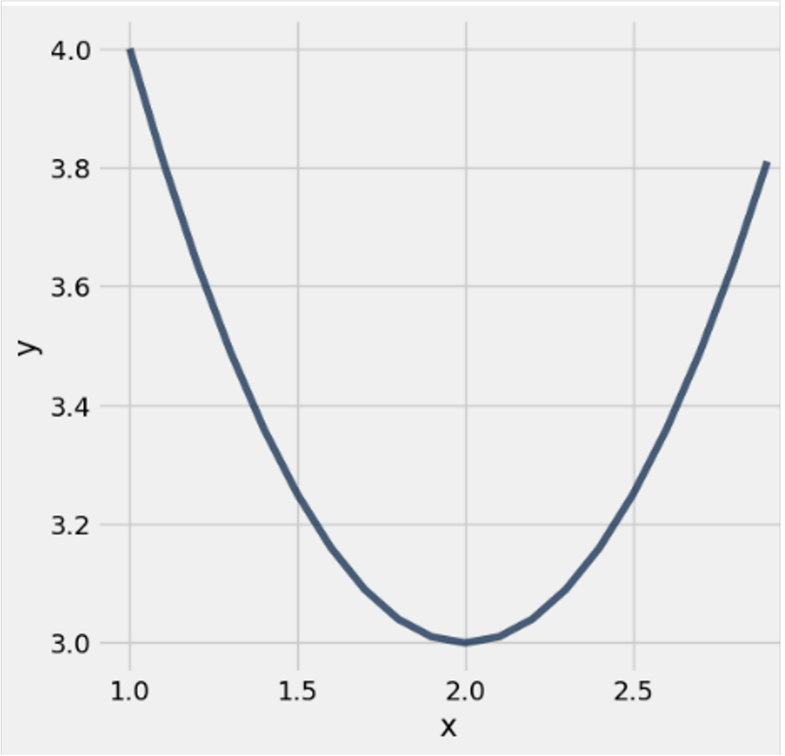
This gives us the **root mean square error** (**rmse**)

Least Squares Line

- **Minimizes** the **root mean squared error** (**rmse**) among all possible lines
  - Equivalently, minimizes the **mean square error** (**mse**) among all lines
- Other names for this line include:
  - “Best fit” line
  - Least squares line
  - Regression line

How to find the minimum?

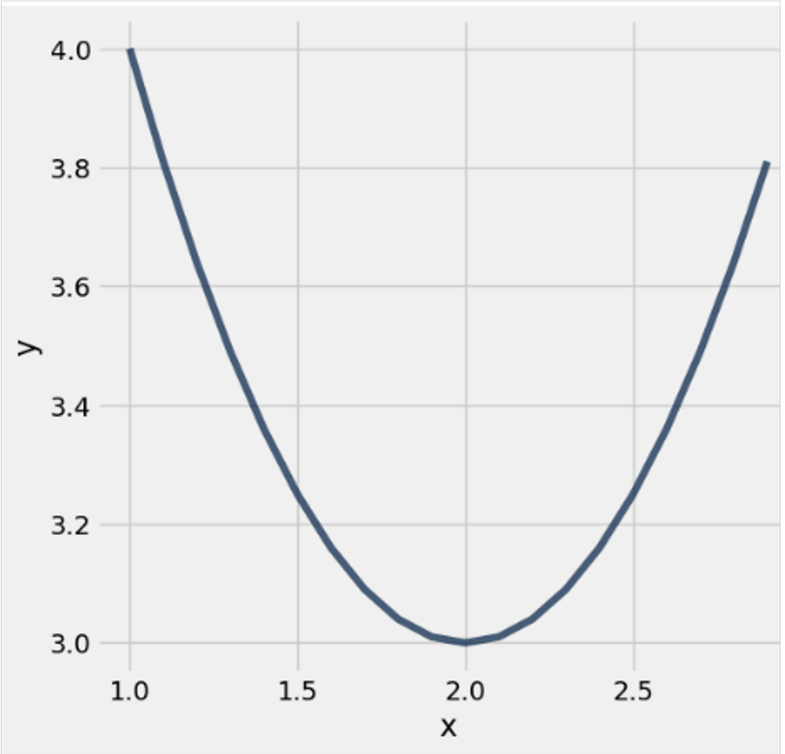
Goal: $\min_x f(x)$



$$f(x) = (x-2)^2 + 3$$

How to find the minimum?

- Derivatives?
  - Find the x where the derivative is closest to zero
  - Only works for simple functions
- Brute force?
  - Evaluate at 1, 2, 3, …
  - This might take a while… How many decimal places should we check?



$$f(x) = (x - 2)^2 + 3$$

Numerical Optimization

- This is a generally difficult task and outside the scope of this class!
- We will use the `datascience` function **`minimize`**
  - **`minimize(f)`** will output **arguments** to a function `f` that minimize the output of `f`

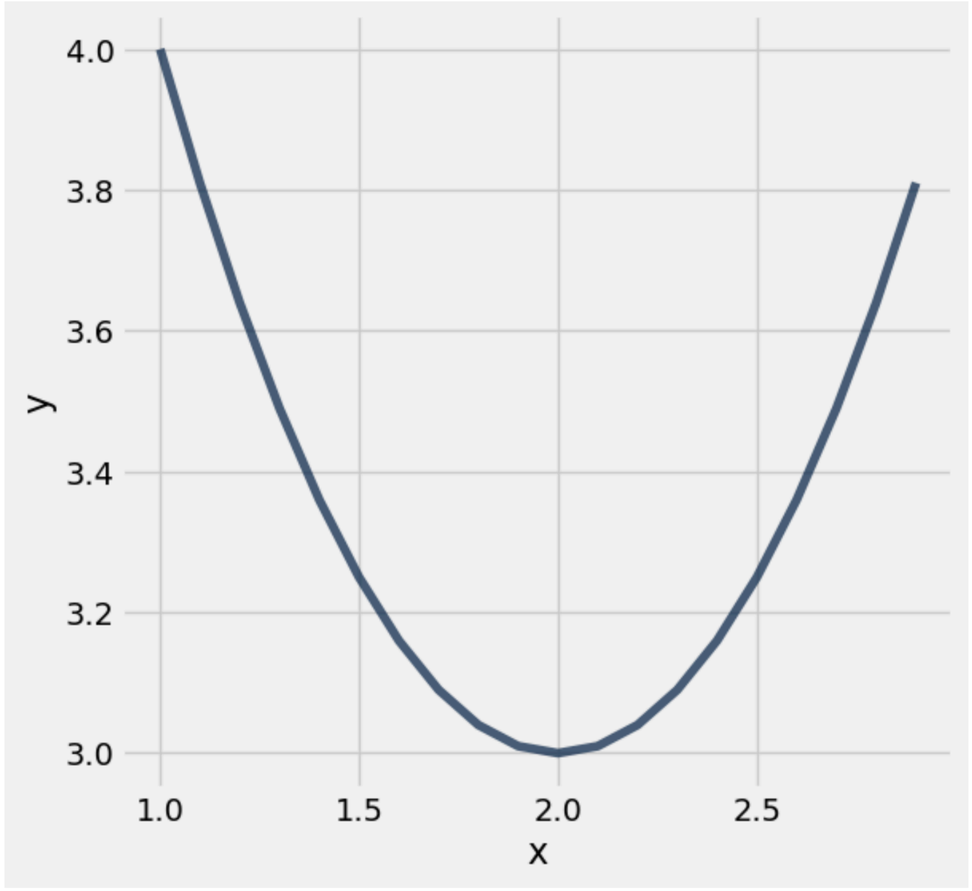
`minimize` Examples

`minimize(f)` will output **arguments** to `f` that minimize the output of `f`

- Suppose we have a function for computing $(x-2)^2+3$:

```
def f(x):
    return ((x-2)**2) + 3
```

- `minimize(f)` outputs 1.9999



`minimize` Examples

- We can also use it for functions with multiple inputs:

```
def f2(x1, x2):
    return 2 * np.sin(x2*np.pi) + x1 ** 3 + x1 ** 4
```

- `minimize(f2)` outputs an array corresponding to `x1` and `x2` that minimize the value of `f2`

```
minimize(f2)
```

```
array([-0.75000601, -0.5       ])
```

Using `minimize` with mse to minimize errors

Suppose we have a dataset with x, y pairs.

If we define a function **`mse(a,b)`** to compute the mean square error of

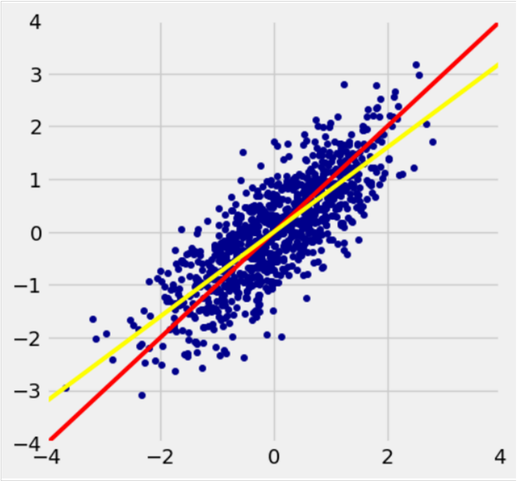
$$\text{estimate of } y = a \times x + b$$

then `minimize(mse)` returns an array $[a_0, b_0]$

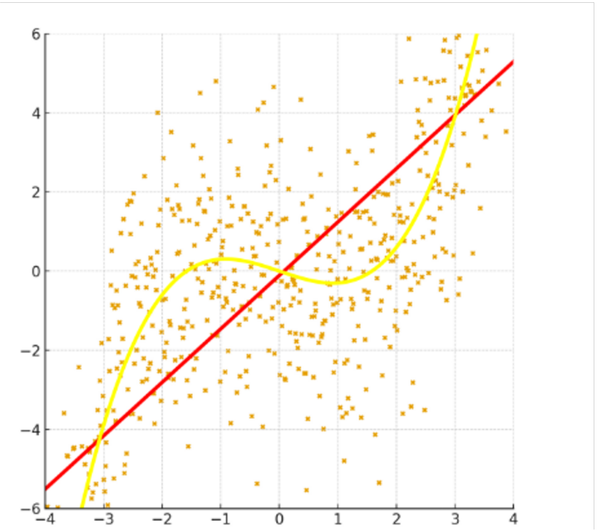
where a_0 is the **slope** and b_0 is the **intercept** that minimizes mse

Least Squares and Residuals

How can we know we've created the best line to fit through our data (i.e., that we've minimized error)?



How can we check whether a line is appropriate (versus a non-linear model)?



Residuals

Residuals

Residual: The error for *individual* regression estimates

$$\text{residual} = \text{observed } y - \text{regression estimate of } y$$

- Can calculate the residual for each individual (x, y) point
- It's the **vertical distance** between the point and the line of best fit

Residual Plots

Plots of residuals can be a diagnostic for whether a linear model is appropriate (versus when a non-linear model might be better)

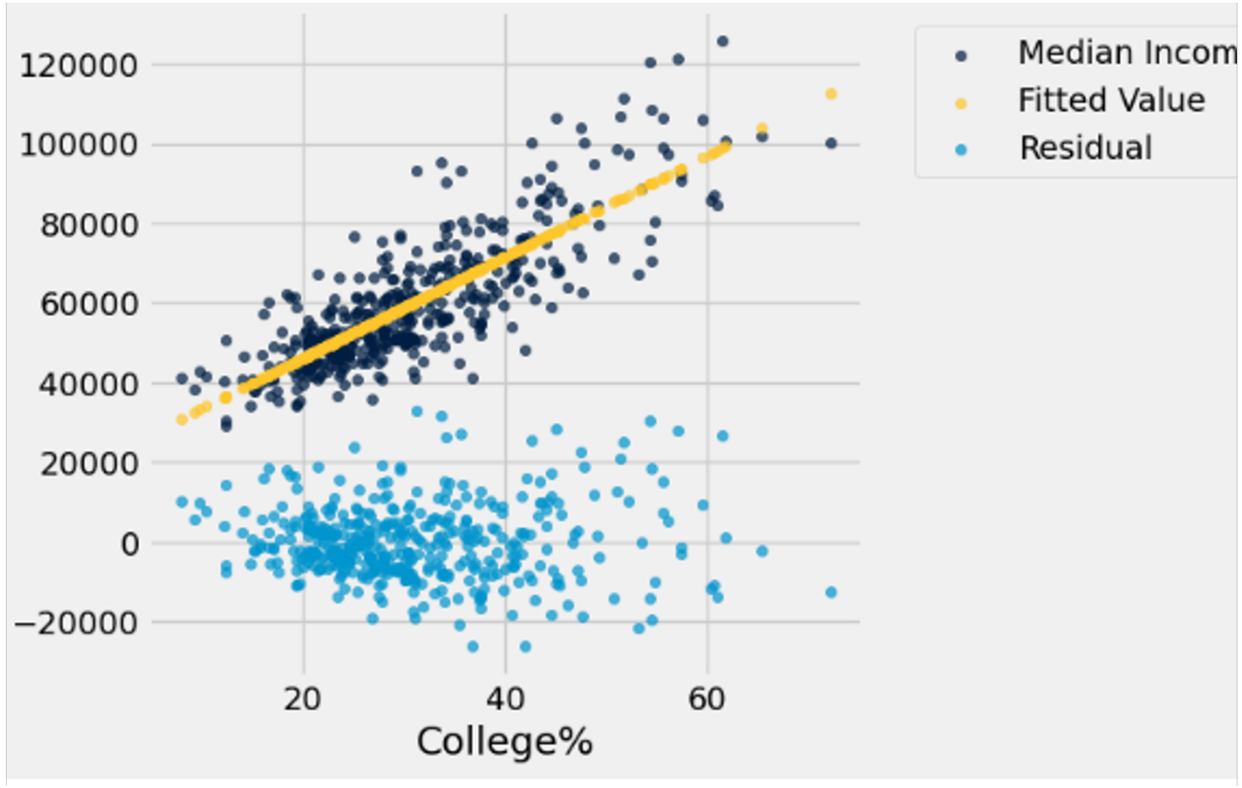
Scatter diagrams of residuals:

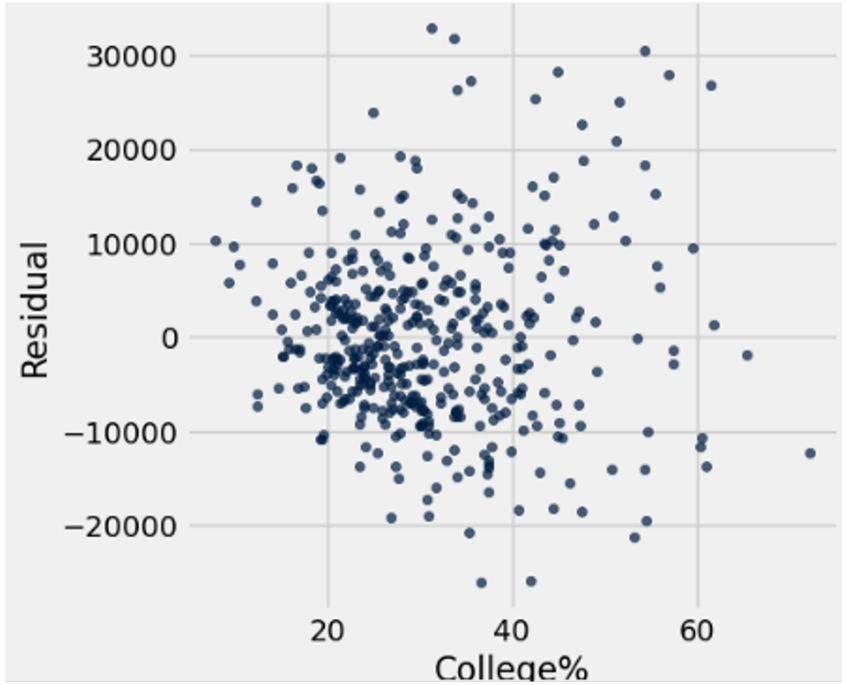
- Should look like unassociated blobs for linear relationships
- Will show patterns for non-linear relationships
- Look for curves, trends, changes in spread, outliers, etc. as examples of non-linear patterns

Properties of residuals

- Residuals from a linear regression always have:
  - Zero mean
  - Zero correlation with x
  - Zero correlation with the fitted values
- These are true no matter what the data looks like
  - Just like how deviations from the mean are zero on average

Residual Plots



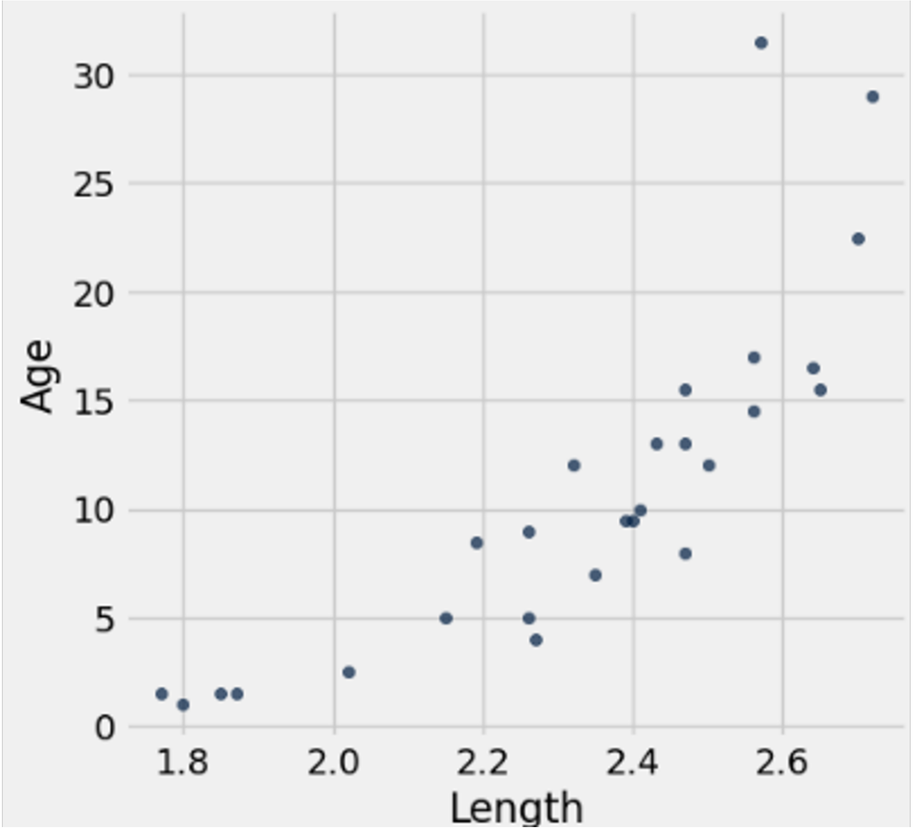


Dugong example

Suppose we have a dataset containing age and length of dugongs

- Is there an association between length and age?

- Is there a *linear* association between length and age?



Notebook Demo: Residuals & Regression Diagnostics

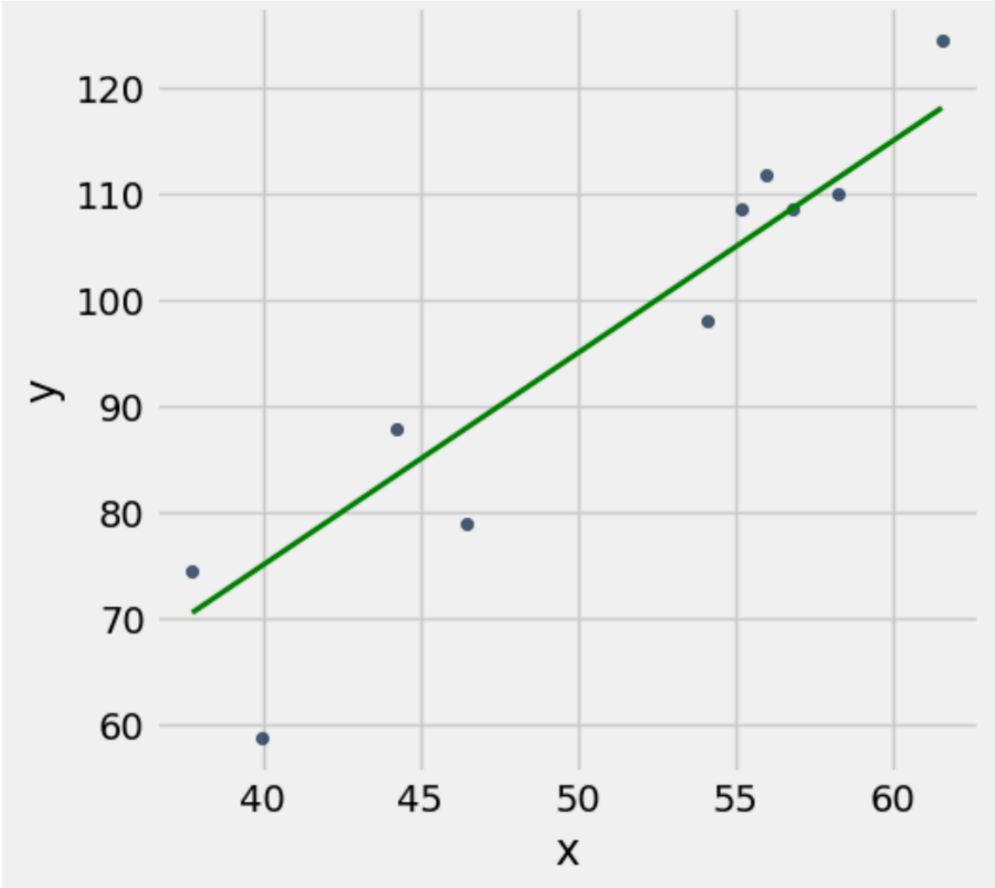
Regression Inference

Regression Inference: Premise

- Our data represents a sample of a larger population
    - The linear relationship (regression line) we determined is dependent on our sample

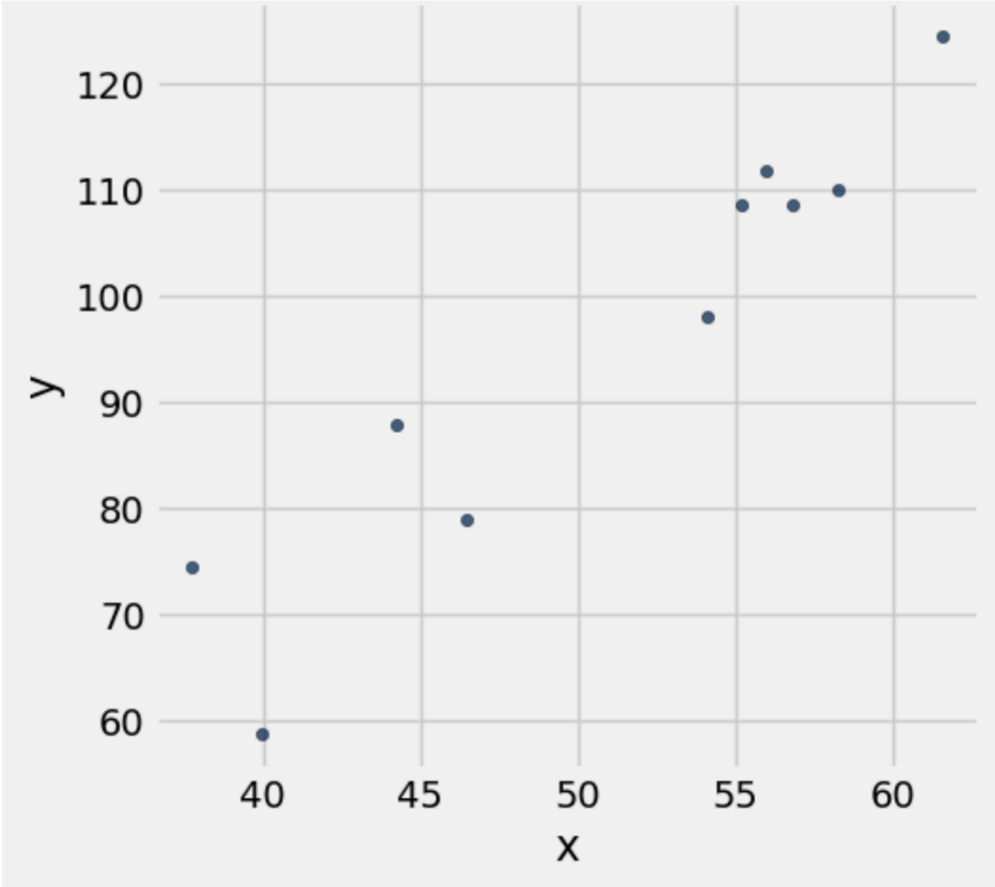
Regression Prediction

True line and 10 samples



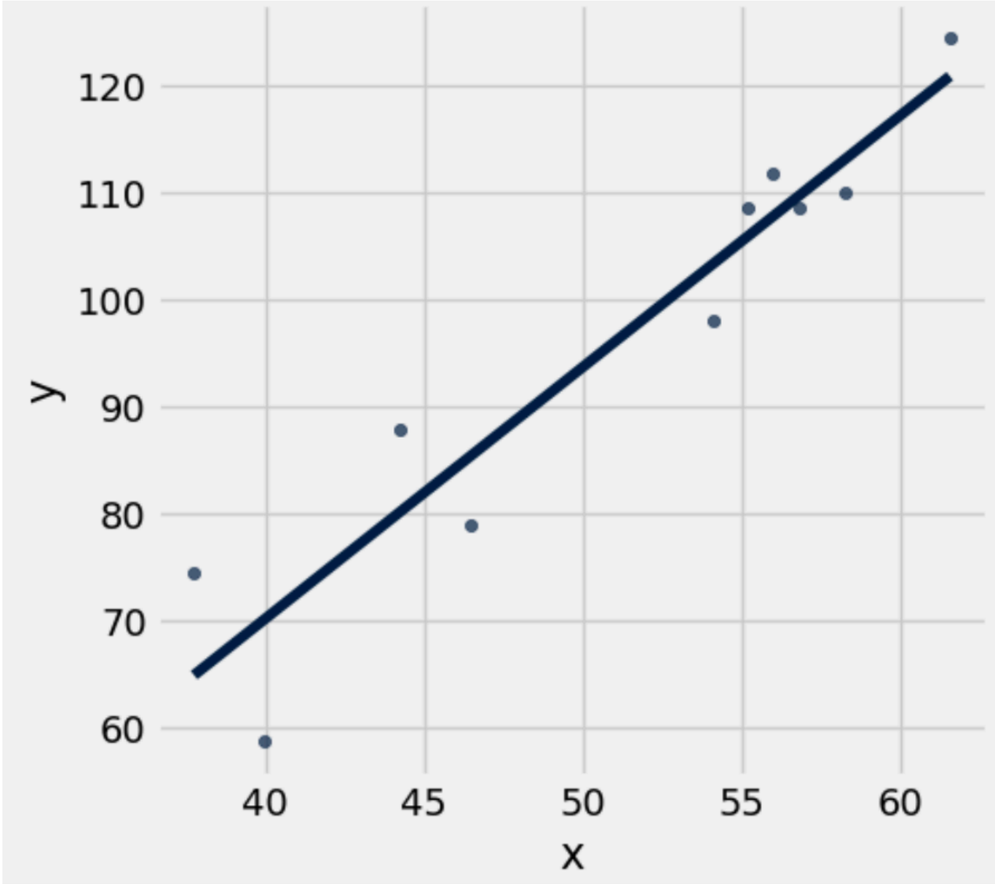
Regression Prediction

What we get to see



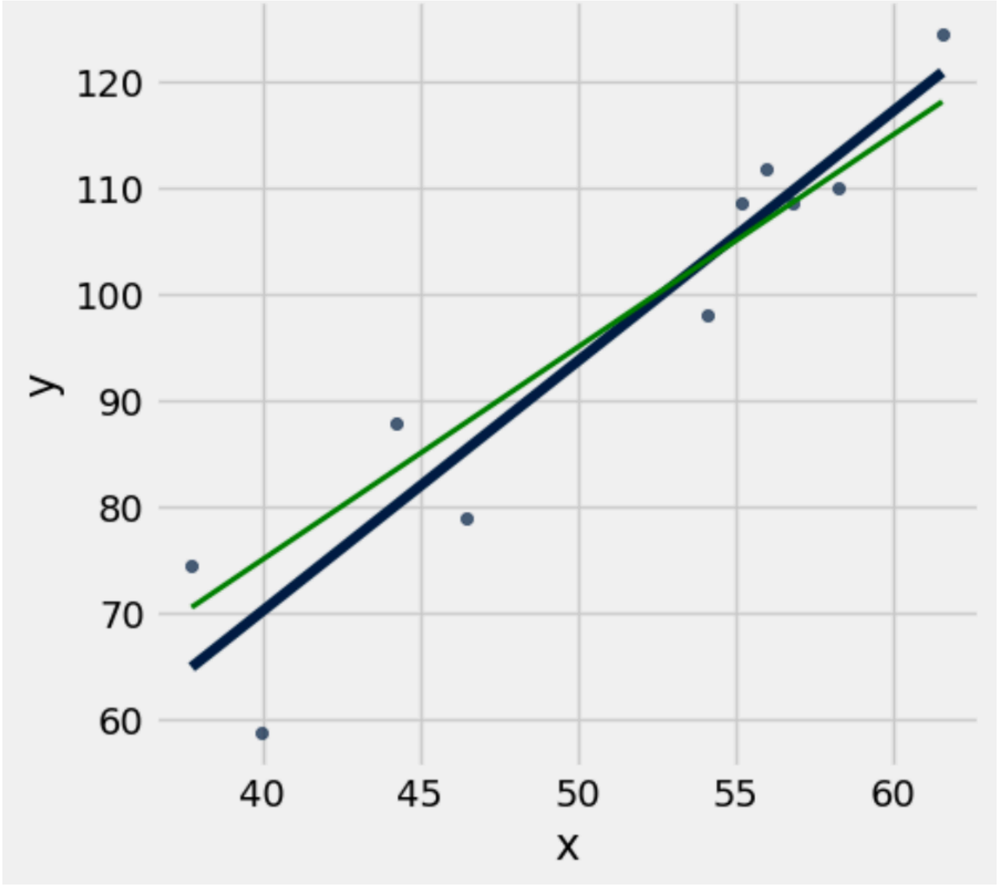
Regression Prediction

Regression Line: Estimate of the True Line



Regression Prediction

Regression Line and True Line



Regression Inference: Premise

- Our data represents a sample of a larger population
    - The linear relationship (regression line) we determined is dependent on our sample
- How confident are we in the regression line that we found?
    - **Estimate uncertainty** with a confidence interval for our **regression prediction**
    - We'll do this using our familiar bootstrap method

Regression Inference: Premise

- **Estimate uncertainty** with a confidence interval for our **regression prediction**

    1. Uncertainty around our **predicted value** for a given x-value

    2. Uncertainty around our regression line **slope**

        - Do we think that the variables are linearly related?

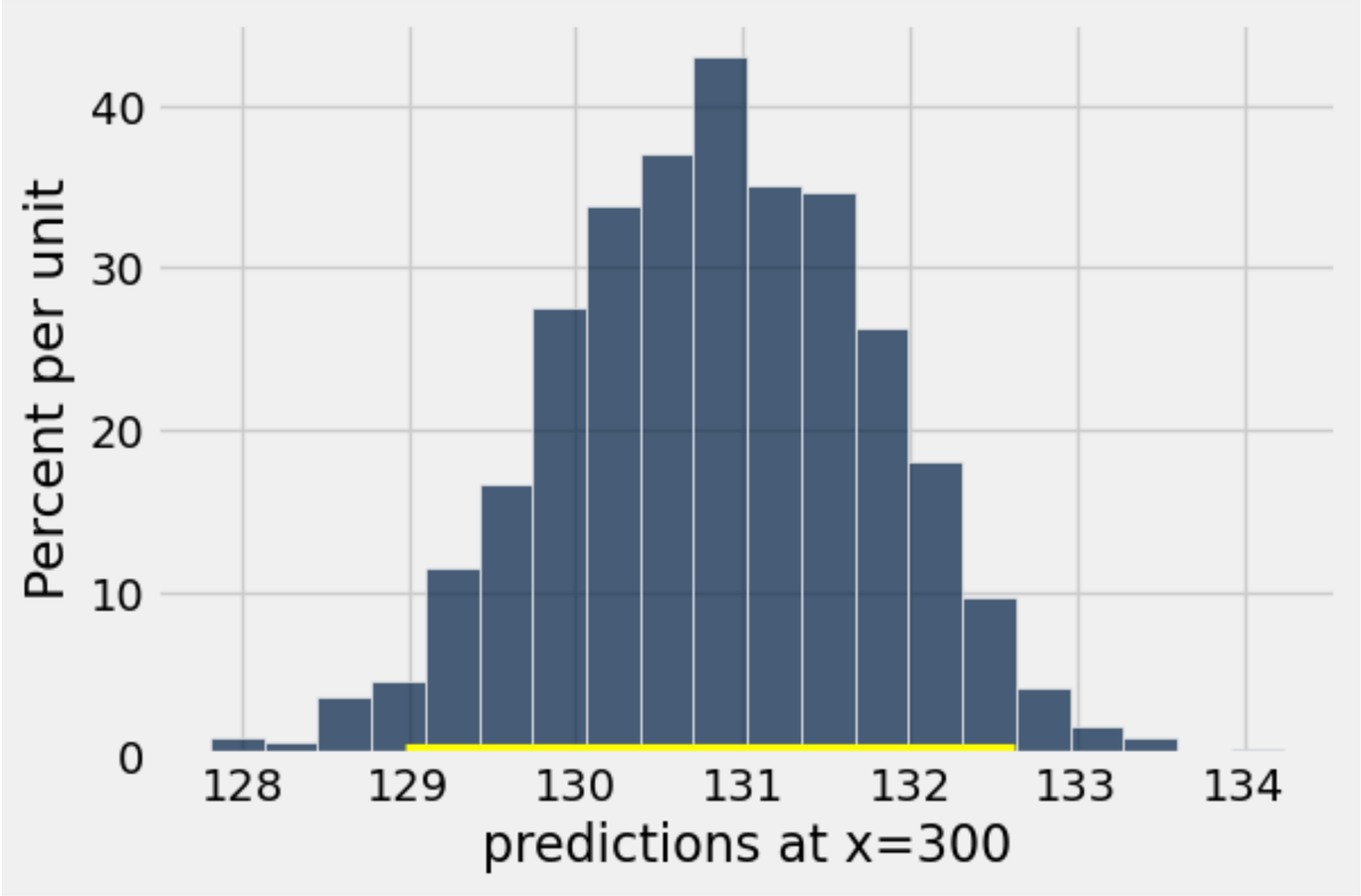
Regression Inference: Bootstrapping

To predict a value y at a given value of x,

we calculate the **predicted value** for each regression line

then draw a **histogram of these predicted values**

and construct our confidence interval



Determining the Prediction Interval

- Bootstrap the scatter plot
- Calculate the regression line for that scatter plot
- Get a prediction for y using the regression line
  - Repeat many times to get a histogram of values for y
- Get the middle 95% interval for a 95% confidence interval
- Interpretation:
  - **95% confidence interval for the height of the true line at x**

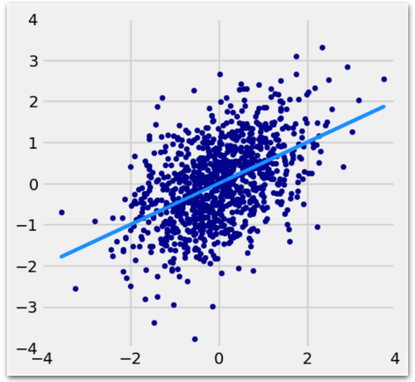
Regression about the True Slope

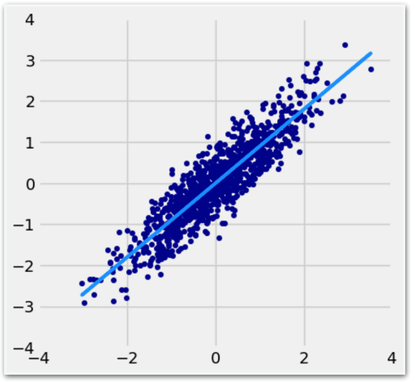
Confidence Interval for Slope

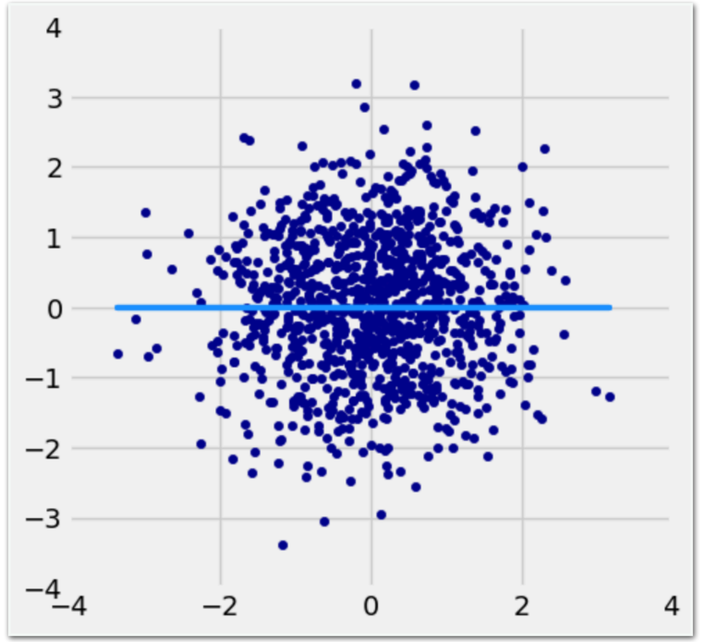
Recall what we know about slope:

- Metric for strength of the linear relationship!

- *When the slope is zero, there's no linear association*







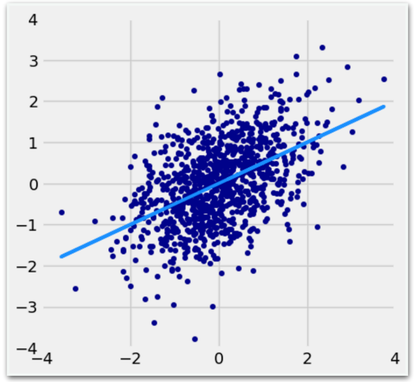
Confidence Interval for Slope

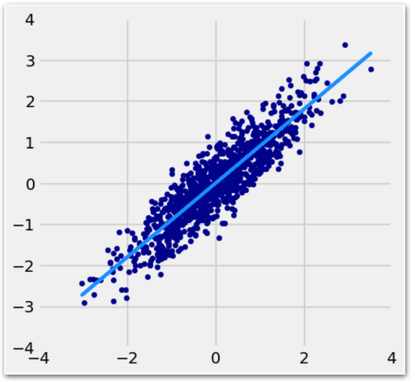
Recall what we know about slope:

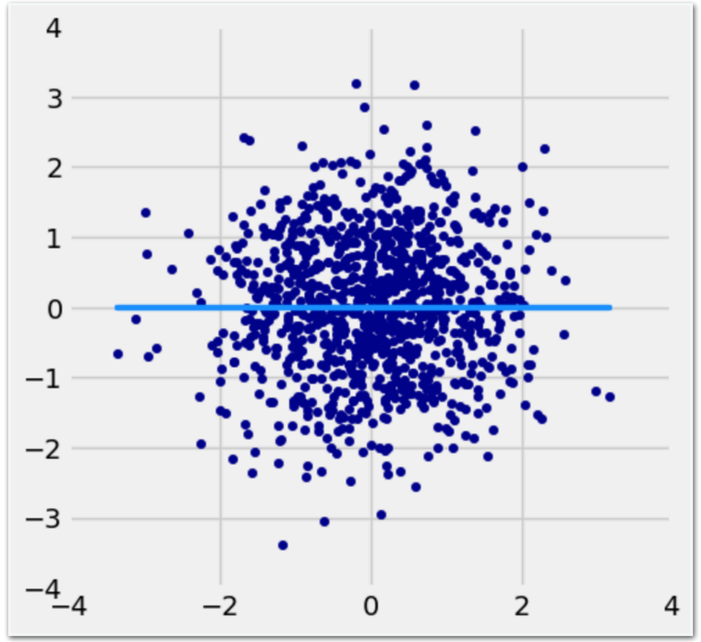
- Metric for strength of the linear relationship!

- *When the slope is zero, there's no linear association*

Same process as we just did for confidence interval of prediction of y at value of x, but instead create a **histogram of slopes**

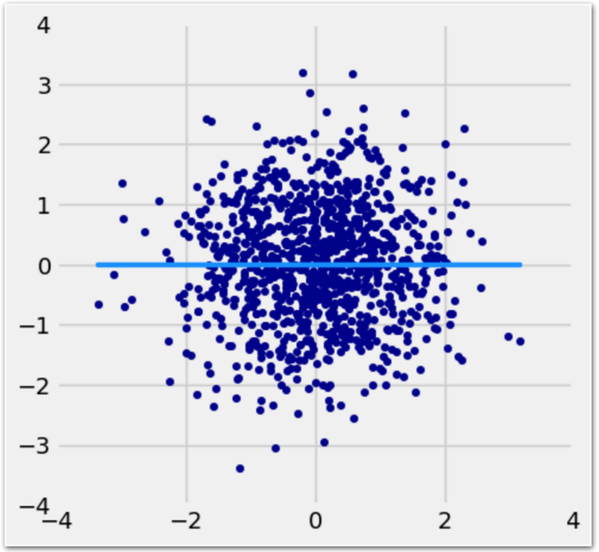




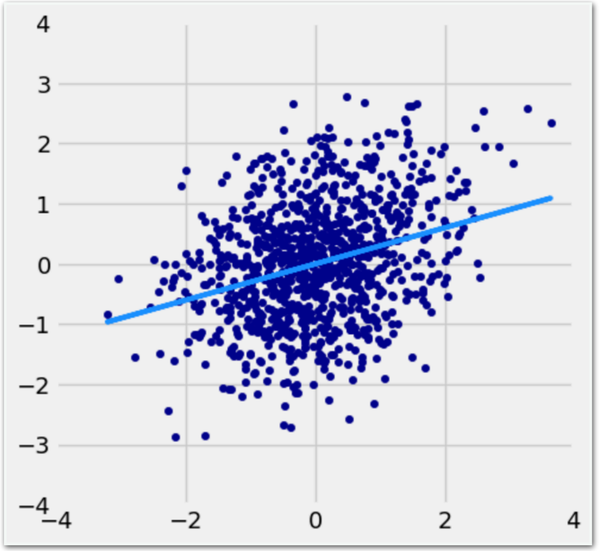


Testing Whether There Really is a Slope

Null Hypothesis:

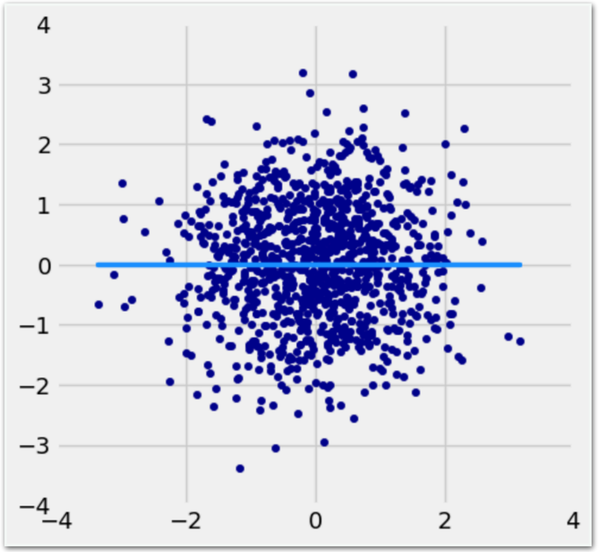


Alternative Hypothesis:

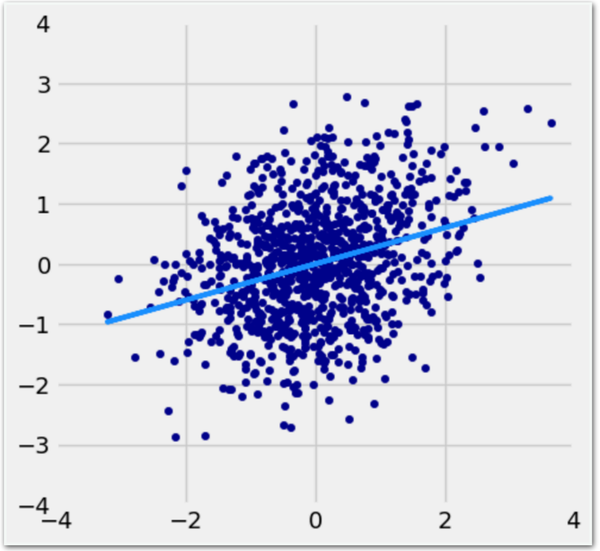


Testing Whether There Really is a Slope

Null Hypothesis: The slope of the true line is 0 (the variables are *not* linearly correlated)



Alternative Hypothesis: The slope of the true line is non-zero



Hypothesis Testing Slope

Null Hypothesis: The slope of the true line is 0 (the variables are not linearly correlated)

Alternative Hypothesis: The slope of the true line is non-zero

- Process:
  - Construct a bootstrap confidence interval for the true slope
  - If the interval doesn't contain 0, **reject null**
  - If the interval does contain 0, **there isn't enough evidence to reject the null**

Next time

- Classification

- Common errors with data