Data 8 Teaching Guide

Oscar Syu, Fall 2017 / Spring 2018 Tutor

May 2018

1 Introduction

About this guide:

This guide serves as a compilation of teaching methods and analogies that address teaching pain points that instructors experience throughout the course. By no means is this guide a model for teaching nor a exhaustive list of topics and teaching methods. Rather it is meant to record and provide support to tutors and GSIs to explain difficult concepts to students. The Teaching Guide's goals are to highlight possible confusion, help accelerate on-boarding for new instructors, provide creative and effective teaching methods, and serve as a living document that records new teaching ideas as Data 8 evolves.

The document is structured in the following format:

- Explanation of Problem
  - Explains the focus of the section and the general approach to teaching the concept
- Problems, Pain Points, and Confusion
  - Frequent areas of confusion and recurring mistakes/patterns with strategies to address them.
  - Each problem is paired with a solution underneath labeled Method [number].
- Points of Emphasis
  - Concepts to stress
- Visuals
  - Graphs, charts, diagrams that illustrate the concept in visual form

Contents

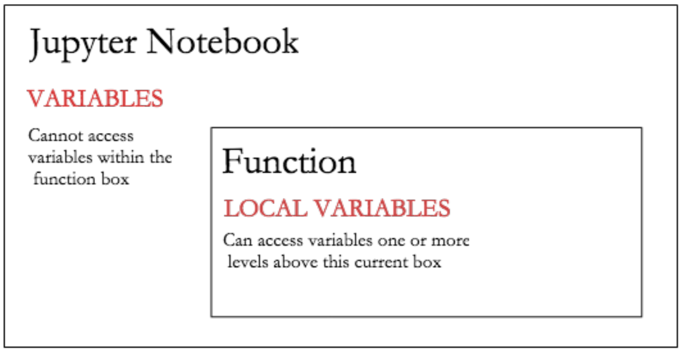
2 Coding

2.1 Functions

- Explanation of Problem
  - The concept of functions may seem foreign to new coders
  - Exists a disconnect between hard-coding and generalization
  - Confusion over what variables are accessible inside and outside the function
- Problems, Pain Points, and Confusion
  - Not understanding the flow of a function
    - Method 1: Refer to below diagram
      - First line declares a name and what parameters to pass
      - Body will sequentially execute each line
      - Return line signifies the end of the current process and gives the user a result

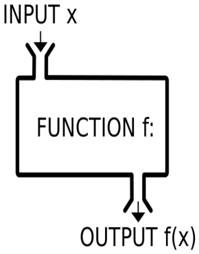
```
def function_name(x, y, z):
    """emphasize naming functions
       and parameters with unique names"""
    <body> #generalizes code, replace w/ x, y, z
    return #return hands the result to the user
```

  - Not understanding the point of a `return` line, or confusing `return` with `print`
    - Method 1:
      - Write two basic functions, one with a `return` line, one without, and one with a `print` line
      - Note how the `return` and `print` functions have an output when called and the one without either has no output
      - Assign variables to calling the `return` and `print` functions, then call the variables and note that one of them is blank. This should demonstrate that only the `return` line ”saves” the result and the `print` line only displays the result for the user.
  - Trouble converting hard-coded work into generalized functions
  - Hard-coding parts of the function where they shouldn’t
  - Unsure of what arguments are necessary to complete a function
    - Method 1: List out all the variables that need to be changed, and replace all the hard-coded numbers with the variables
    - Method 2: Drawing a parallel to math functions is one way to help students understand that def statements operate in the same way in that there is an input x and an output y
      - Think about how you would generalize a math equation like $y = 5 * 2 + 3$ but instead of getting 13 every time, you change the factor multiplied by 5 to get different answers
      - To generalize, you would replace 2 with x and y with f(x) to represent which arguments to pass in: $f(x) = 5x + 3$
  - Using variables defined inside a function (in a local frame) in other places (the global frame or another local frame)
    - Frames and environments are not covered in this course, so instead think about things in terms of boxes:


Jupyter Notebook
VARIABLES
Cannot access variables within the function box
Function
LOCAL VARIABLES
Can access variables one or more levels above this current box

    - This diagram illustrates that variables inside the smaller box are inaccessible by the outside box.

- Points of Emphasis
  - The purpose of functions is to generalize hard-coded actions and act as a flexible option for different situations
  - Avoid using the same variable names or reuse parameters to reduce name confusion
  - Don't necessarily need to generalize everything, just only the parameter of interest.
- Visuals



2.2 For Loops

- Explanation of Problem
  - Students may understand def statements, but for loops may not be intuitive because the structure may seem unfamiliar and the concept of updating a variable is abstract.
  - Easy to miss crucial steps in a for loop because of a lot of moving parts as the loop repeats
- Problems, Pain Points, and Confusion
  - Not seeing difference between using the array in the for loop and using `np.arange(len(array))`
    - Method 1: Write out the logical steps that a for loop goes through to process the array:
    - Given a data array `np.arange(4, 11, 2)`, using a `np.arange(len(array))` method, the array is converted into `np.arange(0, 1, 2)` as an intermediate step
    - But, using the array directly, the for loop will run through `np.arange(4, 11, 2)`
    - When the for loops iterates, it will take the values directly from the array
    - Stress that the **number of times** the for loop iterates over is the same amount regardless, the difference is the values i takes in the array.
    - This difference is useful when applying for loops to problem solving

        - The `for i in np.arange(len(array))` method is useful in situations where consecutive numbers are needed such as enumerating data, or adding accumulating values

          ```
          ex_lst = make_array(15, 12, 57, 29)
          ex_output = make_array()
          for i in np.arange(len(example_lst)):
              ex_final = np.append(ex_final,i)
          ex_final
          >>> array(0, 1, 2, 3)
          ```

        - The for i in array method is useful in situations where an array's data values need to be directly changed, such as calculating tax from a dataset of purchases

          ```
          ex_lst = make_array(15, 12, 57, 29)
          ex_final = make_array()
          for i in example_lst:
              ex_final = np.append(ex_final,i*.0925)
          ex_final
          >>> array (1.3875, 1.101, 5.2725, 2.6825)
          ```

        - Both work when the exact i value does not matter, such as bootstrapping or sampling, unless you want to sample accumulating sample sizes
        - For example, given a bike rental dataset called `bike`

Bike Number	Starting City	Distance Traveled (mi)
483092	Berkeley	10
849204	Oakland	3
105837	San Francisco	25

          ```
          final = make_array()
          for i in np.arange(5000):
              sample_row = table.sample().column(0)
              final = np.append(final,sample_row)
          ```

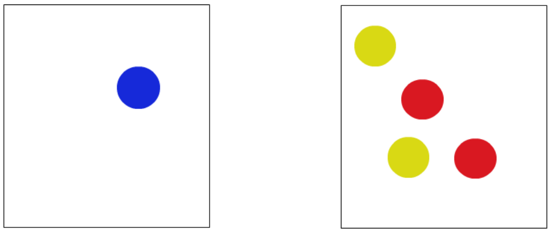
  – Understanding the general flow of a for loop and why initializing array using `make_array()` is necessary

    * Method 1: Think about physically repeating a process over and over again, for example using sorting marbles

        - In real life, there would be two boxes, one with all the marbles (an original data array), and one that is empty where sorted marbles are placed (an empty array set up by the `make_array()` function)
        - Or, the `make_array()` can be thought of as a piece of paper recording marble picks
        - A marble is picked out of the box, inspected, and if it meets the conditions (if/elif/else statements), it is put into the second box (appending the data point to the empty array)
        - Do it for all remaining marbles until there are no more marbles
        - The `return` essentially hands the second box to the user

          ```
          marbles = make_array('blue', 'red', 'yellow', 'red', 'yellow')
          #box with all marbles
          def marble_select(start_data):
              final_array = make_array()
              #second empty box
              for i in marbles:
              #for each individual marble
                  if (i == 'yellow' or i == 'red'):
                  #inspect against conditions
                          final_array = np.append(final_array, i)
                          #put marble in a second box or record it on paper
              return final_array #present box to user
          ```

```
sorted_marbles = marble_select(marbles)
sorted_marbles
>>> array('red', 'yellow', 'red', 'yellow')
```



- Points of Emphasis
  - Don’t use multiple nested for loops unless necessary; it’s just a bad practice and the vast majority of problems will not need them
  - For the purposes of this class, only arrays will be in the for statement.
  - Loops can either repeat the same process over and over again, or build off of the last output

2.3 Table/Array Methods

- Explanation of Problem
  - Especially as the course shifts towards statistics and students don’t have the same practice with table methods as before, they may begin getting confused between table and array methods.
- Problems, Pain Points, and Confusion
  - Easily confused table methods:

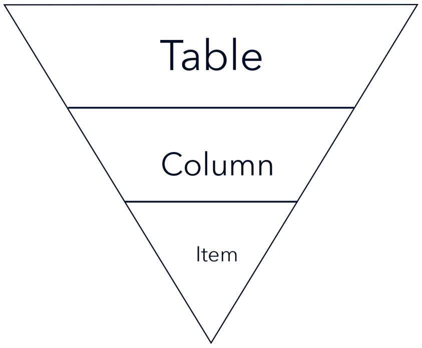
Method	**Differences**
.item, .row	item is applied on arrays, .row is applied on tables, creates a row object, which isn’t really discussed. Students just need to know that it is different from an array
np.random.choice, .sample	np.random.choice is applied on arrays, .sample is applied on tables
.select, .take, .drop	.select creates a copy of a table with specified columns, .take creates a copy of a table with specified rows, .drop creates a copy of a table with specified rows dropped
.scatter, .plot	.scatter shows the distribution of data, .plot draws lines between points to show trends

  - Easily forgotten methods:

Method	**Description**	**Output**
.row	.row(index) returns a row, as a datascience table row, not an array. To make the row an array, use `make_array()`	row object
.item	item(index) selects a value out of an array	selected element
.scatter	scatter(xlabel, `fit_line = True/False`)	graphs
np.diff	finds the differences between adjacent values in an array. The array is 1 less in length than the original	array
.apply	applies a function, either built-in or user-defined on each row. Students may have trouble understanding how to create a function that can be used per row.	array

  - Common mistakes:

        - Question asks for 3 rows of data and students write .show(3). Explain that .show() functions kind of like print(), it is a side effect and displays results
        - Not understanding .pivot
        - Joining tables incorrectly
        - Conceptual problems with grouping
        - Not understanding the granularity of data types



        - Treat the three main data types as its own "world." Perform table methods that output tables until no more can be done. If necessary, then select the column, perform all methods that output arrays before then selecting the item.
        - Example code that follows the flow down the triangle:

          ```
          actors.where("Actor", are.equal to("Tom Hanks")).column("#1 Movie").item(0)
          ```

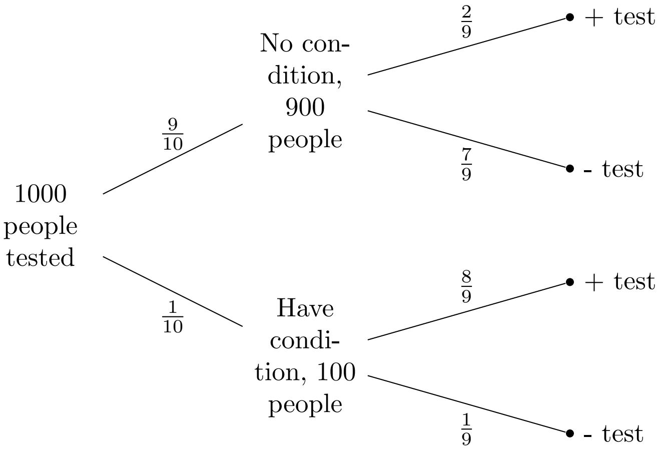
- Points of Emphasis
    - Perform as many operations and drop unnecessary columns as possible before .group because .group will modify other column names. Reduces confusion if .group is the last call.
    - Students may not be sure when to use the np prefix. It's important to stress which ones come from the numpy package.

3 Statistics

3.1 Probability

- Explanation of Problem
    - Probability is one of the trickier topics in that it involves a lot of logic and most of all, practice.
    - At this point in the course, the material moves away from coding to statistics, so the initial change in topic can be jarring
- Problems, Pain Points, and Confusion
    - Confusion between Addition and Multiplication Rule
        - The addition counts the ways an event happens while the multiplication rule is used when events happen simultaneously or dependent on each other
        - Method 1: Think about the two in terms of "or" for addition rule and "and/then" for multiplication rule. Those key words in word problems give hints on how to approach the problem.
        - Confusion about the likelihood an event ends up on the xth pick. (ex probability of an ace of hearts on the 2nd draw with replacement).
            - It is equally likely for every event to be at any xth pick. Hence the probability should still be the same for all of them.

        - Conditional probability: Bayes' Rule Explanation
            - Method 1: Teach Bayes' Rule as a more specific version of the multiplication rule. $P(B|A)$ is the probability of A happening first, then probability of B happening after A happens. Multiply them as though they are two simultaneous events
            - Method 2: One way to remember Bayes' Rule is to use the form $P(A)P(B|A) = P(B) * P(A|B)$ and remember an alternating ABABAB pattern
        - Conditional probability: Bayes' Rule vs non-Bayes' Rule situations
        - Students may have difficulty understanding when to use Bayes' Rule
        - Method 1: Tree diagrams are especially helpful
        - In the following example



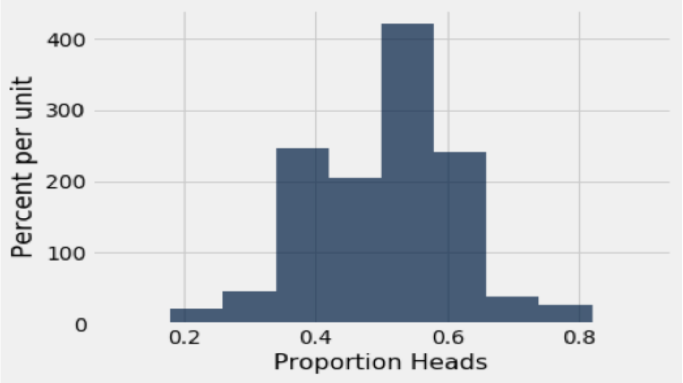
    - In a tree diagram, there exists prior probabilities, given by the first set of branches from the left, and likelihoods by the second set of branches
    - There's always confusion for when the Bayes' rule formula is applicable, especially in cases where there is a given probability
    - If the prior probability is the given probability, the total probability of the specified event will be the likelihood, since there is only one path through the tree diagram
    - For example, in the diagram above, if we were to find the probability of a negative test result given the sample of the 900 people without the condition, the answer would be $\frac{7}{9}$. Essentially, the ”no condition” branch constrains the possible results to + test and - test. Since we are 100% certain that we are only dealing with people without the condition, the probability of a negative result is $\frac{7}{9}$.
    - If the likelihood is the given probability, then there may be multiple paths to the likelihood since there may be more than 1 prior probability that leads to the same likelihood
    - For example, in the diagram above, if we were to find the probability of having a condition given a positive test result, we would have two paths to positive results: No condition/+ test and condition/- test. Essentially, the ”population” of probabilities have been reduced to $\frac{9}{10} * \frac{2}{9}$ and $\frac{1}{10} * \frac{8}{9}$. But, we only want the probability of a + test and has condition $P(+test \cap condition)$. So in this case we use the Bayes' rule formula as we want to find the probability of choosing the desired branch over the possible branches constrained by the given probability. So the formula becomes $P(condition| + test) = \frac{P(+test \cap condition)}{P(+test)}$ or $P(condition| + test) = \frac{\frac{1}{10} * \frac{8}{9}}{\frac{9}{10} * \frac{7}{9} + \frac{1}{10} * \frac{8}{9}}$
    - Method 1: One way to reduce confusion is to circle the branches to figure out the denominator of the Bayes' Rule formula. In the example outlined above, all branch paths to + test would be circled. Then, using the addition rule, the two probabilities are added together to find the specified branch's probability given a set of paths to + test.

- Points of Emphasis

- General formula for probability:
  $P(eventhappens) = \frac{numberoutcomesofevent}{numberofoutcomes}$
- Dice games are always with replacement, Card games can be both
- Tree diagrams are an important visual tool, practice as many as possible.
- Use the $P(eventdoesn'thappen) = 1 - P(eventhappens)$ rule when possible.

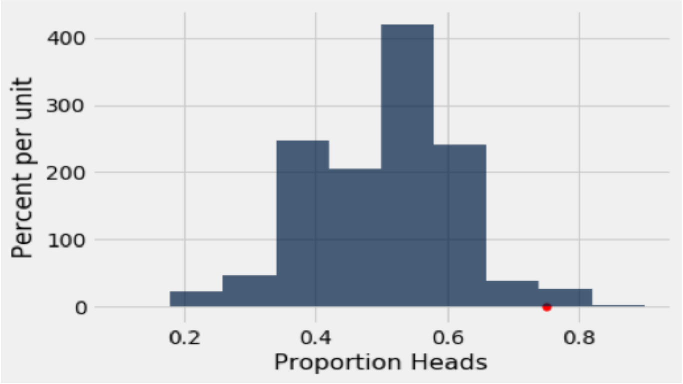
3.2 P-values & Testing Hypotheses

- Explanation of Problem
  - Students may have trouble grasping what exactly a p-value means, why an arbitrary value has an effect on the conclusion of a hypothesis test, and graphically what it looks like.
- Problems, Pain Points, and Confusion
  - Method 1: An effective way of explaining the p-value is to provide a breakdown of the definition, and apply it to some simple example scenario. Try as follows:
    "(3) Probability of an outcome as extreme or more extreme in the direction of the alternative (2) than the observed test statistic, (1) under the assumption that the null hypothesis is true."
  - Example Scenario: Alice and Bob play a game where the winner is whoever gets the most heads in 20 coins flips. Bob goes first and gets a total of 15 heads. Alice suspects Bob is cheating, and that his coin has a higher probability of heads than a fair coin.
  - Set the following hypotheses:
    Null hypothesis: fair coin, expect .5 probability of heads/tails and any variation is due to chance.
    Alternative hypothesis: biased coin, expect greater than .5 probability of heads.
  - Interpret the definition of the P-value more easily by splitting up into 3 components:
    1. ”under the assumption that the null hypothesis is true”
       * This is our first step in breaking down the definition of the P-value. Interpret this statement by simulating the scenario under the assumptions of the null hypothesis. In other words, simulate Bob's 20 flips over a large number of trials given that his coin is fair. To help the student gain a visual interpretation of the p-value, draw out the distribution of the null simulation:

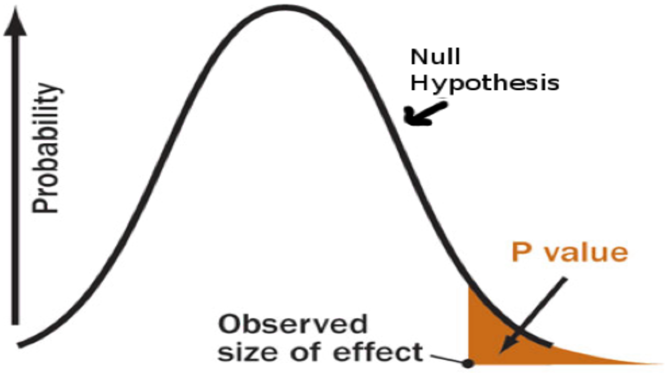


    2. 'the observed test statistic'
       * Plotting the observed test statistic allows us to see how our observed value compares with the rest of the simulated data. In this case, plot the point corresponding to Bob (15/20). Note in this example that there is a relatively low density of values (small bar height) at the plotted point, which indicates that it is relatively uncommon for a proportion such

as this to occur under the null.



3. 'Probability of an outcome as extreme or more extreme in the direction of the alternative'
   - ∗ Point out that the area of some region of the curve out of the total area is equivalent to the probability of getting a value within that region. Then, the probability of an outcome at least as extreme as the value plotted (the observed test statistic) is simply area starting at that point and extending in the direction of the alternative. For this example, it will be helpful to shade in the area of the curve to the right of the plotted point, and write out the equivalent statement in code. This is how to calculate the P-value!
   - ∗ The following graphic summarizes this idea:



- Points of Emphasis
  - Why use $\geq$
    - ∗ Intuition is that we want to see the distribution of similar or more extreme values. If a value is exactly the same as the observed value, that would be equivalent to finding a data point that matches the observed outcome. We then want to test if this outcome is due to chance or not.
    - ∗ One way to think about it is trying to see the values that are "crazier", unlikely or more out of this world than the observed outcome. Being equal to the observed outcome supports its case for validity.
    - ∗ Essentially, hypothesis testing is a form of **proof by contradiction**. The desire is to prove to a certain degree of sureness that the alternative hypothesis holds or if that result cannot be shown, state that null hypothesis cannot be rejected.
  - α Significance level
    - ∗ If the p-value less than α, reject the null hypothesis. α was chosen to be some small value, so that if the p-value is even smaller than that, then it tells us that the chance of observing such an extreme outcome is very likely given the conditions assumed by the null hypothesis. Since the null doesn't describe the outcome, the null is rejected.

        - If the p-value is $\geq \alpha$, we **fail to reject** the null hypothesis. We don't accept the null to be true; instead we demonstrate that our alternative hypothesis is not a better explanation of the data than the null hypothesis. We don't show that the null is correct, rather we show that the alternative does not explain the observed outcome.
        - The null and alternative hypothesis are **mutually exclusive**, so showing one situation does not hold true gives weight to the other hypothesis. Both possible conclusions (alternative and null) made from a hypothesis test are not set in stone. The difference is that the null hypothesis can only be "failed to be rejected" while the alternative is based upon a set confidence level. There is always an error (the α) that could prove to be true.
    - The mean of a binonimal is the proportion.

3.3 Writing Hypotheses

- Explanation of Problem
    - Students may be confused about how to write null & alternative hypotheses. Cases include figuring out the difference between the two, the level of detail required, and the conclusion at the end.
- Problems, Pain Points, and Confusion
    - Difference between alternative & null hypotheses
        - One explanation is that the alternative hypothesis is what the researcher is trying to prove. The null hypothesis is then the exact converse of that statement. Both must be mutually exclusive.
        - In the Bob/Alice example from Section 3.2, the hypotheses would be:
          Null: Bob isn't cheating, and his probability of getting heads is 0.5. We expect him to flip $\frac{10}{20}$ heads and any variation in the outcome is due to random chance.
          Alternative: Bob is cheating, and his probability of getting heads is greater than 0.5.
          In this case, we are trying to determine if Bob is cheating, so that statement becomes the alternative hypothesis. Then the converse is that Bob is not cheating.
        - One generalization in determining the null hypothesis is that many times, it is the one that is the result of randomization as a hypothesis test attempts to establish a baseline condition to show a contradiction with the alternative hypothesis. But, be careful with this rule as there may be exceptions.
    - Level of detail in writing hypotheses
        - More specificity is always better. Include context and situation when writing them.
        - For the null hypothesis, many times it is better to include "any variation in the outcome is due to random chance" if appropriate to make the difference between null & alternative clearer.
    - Rejecting/failing to reject a null hypothesis
        - A null hypothesis should never be accepted, as it is the a baseline condition that states that there is not enough evidence to prove the alternative. We don't accept the null to be true; instead we demonstrate that our alternative hypothesis is not a better explanation of the data than the null hypothesis. We don't show that the null is correct, rather we show that the alternative does not explain the observed outcome.
        - Method 1: Scientific Analogy. In the case of science, the hypothesis is something we try to support with evidence or outright prove is wrong. In the same way a scientific theory is called a theory because it cannot be 100% proved, the null hypothesis follows that rule in that it can only be refuted, but never proved.
        - Method 2: Law Analogy. The phrase "Innocent until proven guilty" has parallels to "Null until alternative." There is no way to make any judgment on a condition until there is enough evidence to support the alternative conclusion

        - Method 3: Narrative Analogy. In a very simplified case of proving the existence of extraterrestrial life, the hypotheses would be:
          Null: Aliens do not exist
          Alternative: Aliens do exist
          If an alien was seen visiting Earth, there would be enough evidence to prove that aliens do exist (the alternative). Remember, however, this is a very simplified case so in most hypothesis testing there is still a level of uncertainty with the introduction of αs. But if an alien was never observed, there is no way to prove that aliens do not exist as the universe is too vast. The only statement that rings true is that **there is not enough evidence to prove the existence of aliens.**

- Points of Emphasis
    - A rigorous hypothesis should contextualize the given scenario, as well as indicate a numeric value for the test statistic under the null/alternative.

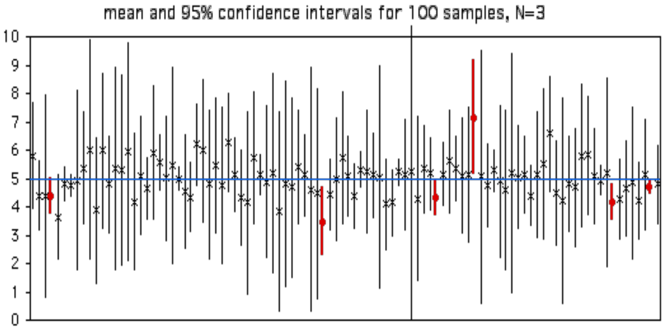
3.4 Confidence Intervals

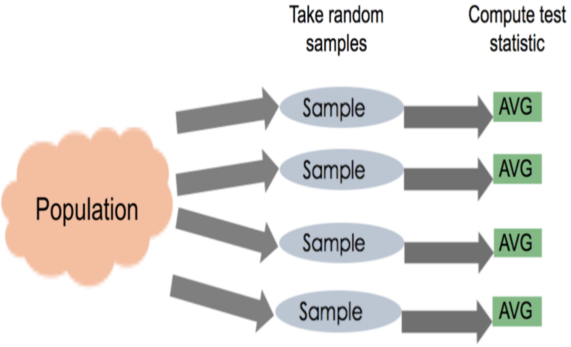
- Explanation of Problem
    - Confidence intervals are tricky because of the wording, what they represent, interpretation, and their relationship to p-values
- Problems, Pain Points, and Confusion
    - Students may be confused about what situations require a confidence interval
        - Confidence intervals are used in cases of sampling (including bootstrapping) in which their is uncertainty about the parameter. When there is no access to the full population, confidence intervals become necessary to give an idea of the range the parameter should fall.
        - Two ways to create confidence intervals: Take more samples from the population or bootstrap
    - Students may have trouble understanding what a confidence interval can predict.
        - Confidence intervals do not measure the chance the parameter is in the interval
        - Rather it measures how certain a confidence interval contains the true parameter
            - Method 1 Coin Flip Analogy: - Before we flip a coin, there is a 50-50 chance of being head or tails. Once we flip the coin, it is either heads or tails, there is no more chance. Similarly, before we compute our confidence interval, 95% of the confidence intervals we compute will contain the true parameter. Once we have computed our CI, there is no more chance; it either contains the true parameter or does not.
            - The parameter is a measure of the population and does not change during the estimation process.
    - It may be difficult to understand what 0 in a confidence interval means
        - This situation shows up when trying to test the difference between two data sets.
        - 0 tells us that we can't make a determination comparing the 2 data sets.
    - It may be hard to understand the relationship to p-values
        - One way to think about p-values and confidence intervals is that the p-value is that it is an error rate and that with 95% confidence intervals, we expect 5% of the time to get intervals that do not capture the population parameter.
    - Proper wording of confidence interval analysis
        - The correct statement for confidence intervals is "We are 95% confident that the true population ¡insert parameter¿ is between (lower bound, upper bound)."
            - The term "confidence" denotes a certainty and represents the idea that either the population parameter is in the interval or not. This is distinct from chance in that chance denotes a probability.

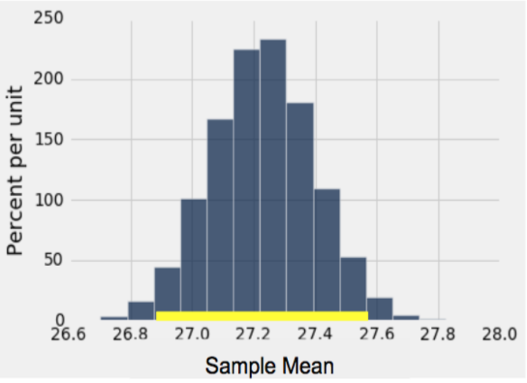
        - ∗ Incorrect statements may include ”There is a 95% chance that the true population ¡insert parameter¿ is between (lower bound, upper bound).”

- Points of Emphasis
    - Review what a parameter is, and state that it does not change. Many students lose sight of this, but the parameter is a measure of the population that does not change during our estimation process
    - We define the CI by the left endpoitn and the right endpoint. ex the 2.5th percentile and the 97.5 percentile for a 95% confidence interval
    - This process gives us a 95% CI where we can say we are 95% confident the true parameter lies in this range. This means if we were to repeat the process of computing our confidence interval, we would expect 95% of those intervals to contain the true parameter.
    - It is introduced around the same time as bootstrap. CI is not necessarily computed with bootstrapped samples, but often is
    - Used with hypothesis testing, usually to determine if a given CI contains 0 or not. If it does, this usually means we fail to reject the null since the null (usually) is the parameter is 0 any and variance from that is due to chance. The confidence interval demonstrates the variance due to chance
    - A wrong statement that appears often from students: “There is a 95% probability that the population parameter is in the interval”. The interval will either contain the population parameter, or it won’t. The interval is fixed (decided), as is the population parameter, so there is no involvement of chances. The “95%” part does not describe the accuracy of the interval, it describes the accuracy of the PROCESS that generates the interval.

- Visuals







3.5 Bootstrapping

- Explanation of Problem

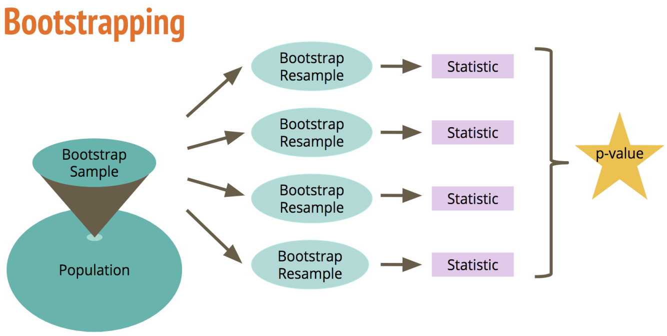
  - Bootstrapping is fundamental to many topics moving forward. Students may not understand the purpose of repeatedly picking from the same sample.

- Problems, Pain Points, and Confusion
  - Why can we not just keep sampling from the population?
    - We often do not have access to the population in its entirely. Ideally, we would SRS. But instead we can only treat our single sample as a population to randomly pick from.
    - Method 1: Pretend that you're a poor graduate student trying to do some research project. Your task is simple, estimate the average age of all people in the US. Does it seem feasible to carry out the method we reviewed earlier? To put it into perspective, if we're taking 500 samples of size 1000, that means you have to go out and find 1000 random people... 500 times. Not. Feasible. You'll have the time/money/resources to gather up one random sample of 1000 people – that seems feasible. After that, that's all the data you'll have to work with.
  - Why does this work?
    - Our sample isn't as large as the population, but we assume it's representative enough. Think back to the Law of Averages: large samples are indicative of the population they were drawn from.
  - Steps of the bootstrap method:
    - Treat the original sample as if it were the population
    - Draw from the sample, at random with replacement, the same number of times as the original sample size
    - This is the same process as above, except we cannot draw our samples from the population anymore. Instead, we insert in a step in between. You sample ONCE from the population and call this your "original sample". Then you create all your new samples from this original sample, also known as the bootstrap method.
    - Example code:

      ```
      def bootstrap_ex(sample_table, trial):
          samples = make_array() #array of all test_statistics
          for i in np.arange(trials):
              bootstrap_sample = sample_table.sample() #each sample is the same size
              bootstrap_mean = np.mean(boostrap_sample.column(0)) #calculate mean
              samples = np.append(samples, bootstrap_mean) #append ot samples
          return samples
      ```

- Points of Emphasis
  - Bootstrapping is the next best option to SRS and works surprisingly well
  - Memorize the flow of bootstrapping because many other concepts use the same format, ex regression inference
  - The result of the bootstrap method is a bootstrap empirical distribution for the sample statistic that estimates the population parameter.
  - Next step is to use confidence intervals and pick the middle 95% of the interval of the estimates

- Visuals



3.6 Correlation

- Explanation of Problem
  - Conceptually the idea of correlation is not too hard to grasp: if a and b are related, you can probably see some sort of a relation and you can measure it with some coefficient
  - The confusion comes from calculating it and how to apply the concepts
- Problems, Pain Points, and Confusion
  - Why is r calculated the way it is?
    - This is beyond the scope of this class but to explain conceptually, r basically compares the distances of the x and y arrays relative to each other using standard units. It compares variability of each dataset against each other.
  - Can r be used to describe nonlinear relationships
    - No, r is exclusive for linear regression
- Points of Emphasis
  - r is **not** the slope of the linear regression line
  - Switching x and y when calculating r does not have an effect since we are averaging a product. Conceptually, two variables have the same correlation to each other
  - r has no units
  - -1 means a strong negative correlation, not no correlation
  - **correlation does not imply causation**
    - Real World Example: Vaccines and autism are associated, but not casual. The timing of some vaccinations in infants coincides with the timing autistic symptoms become observable so it may seem like vaccinations cause autism, but in reality time is the confounding factor.

3.7 Linear Regression

- Explanation of Problem
  - Linear regression is the only model we focus on in Data 8
  - Problems usually arise with conceptual understanding
- Problems, Pain Points, and Confusion

- Difference between the standard units and original units equation. Proof they are the same equation below:
  standard units equation:

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

  original units equation:

  $$y_{est} = r \times \frac{SD_y}{SD_x} x + intercept$$

  where

  $$intercept = \bar{y} - slope \times \bar{x}$$
  $$slope = r \times \frac{SD_y}{SD_x}$$

  definition of standard units of x and y

  $$x_{su} = \frac{x_{obs} - \bar{x}}{SD_x}, y_{su} = \frac{y_{est} - \bar{y}}{SD_y}$$

  plug in definitions of SU into standard units equation

  $$\frac{y_{est} - \bar{y}}{SD_y} = r \times \frac{x_{obs} - \bar{x}}{SD_x}$$

  multiply both sides by SD_y and move y_bar to the right

  $$y_{est} = r \times \frac{SD_y}{SD_x}(x_{obs} - \bar{x}) + \bar{y}$$

  replace $r \times \frac{SD_y}{SD_x}$ with $slope = m$

  $$y_{est} = mx_{obs} - m\bar{x} + \bar{y}$$

  $\bar{y} - m\bar{x}$ is the formula for the intercept of the original units equation

  $$y_{est} = slope \times x_{obs} + intercept$$

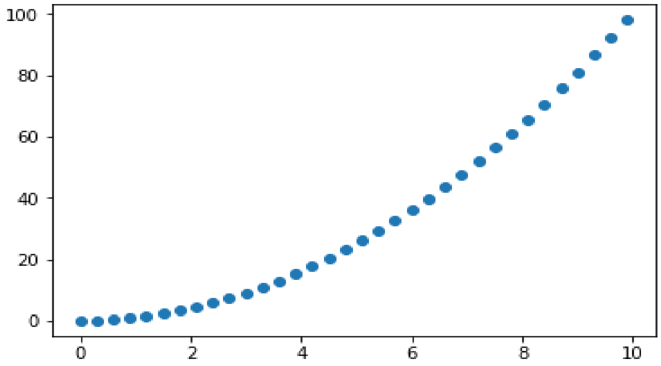
  $$Q.E.D$$

- What conceptually is a linear regression line?
  - In its simplest terms, is taking what data is available, and draw a line in the middle of the data and extending past the data.
  - Given an input x value, will find a corresponding y value that lies on the linear regression line.
- Why do we take the square root in the RMSE?
  - The RMSE is the stndard deviation of residuals. We square residuals to avoid cancellation, then square root to put back into units we can understand.
- Standard units are how far from the average of distribution a point is. Wtih the average being 0 units away from the average.

- Points of Emphasis
  - Linear regression is literally just a line, which is both powerful and has ramifications
  - Because it is only a line, we can mathematically predict values very far from the data cluster, but it doesn't account for other patterns or behaviors.

    - Real World Example: In economics, production costs tend to fall as quantity increases, known as economies of scale. At a certain point, however, it enters period of constant costs and then because of inefficiencies, rise again.
    - If given only data for production in the region of lowering costs, a linear regression model would assume costs would decrease infinitely when in reality it does not
    - The correlation coefficient is what allows us to define the relationship between x and y and find a predicted value.

- Visuals
  - Below is an example for which linear regression wouldn't be a great fit.



3.8 Residuals

- Explanation of Problem
  - Connection between residual plots and linear regression models can be unclear
- Problems, Pain Points, and Confusion
  - What is a residual plot?
    - It is essentially the plot of all residuals, or the distances between the observed and predicted points. Treating the regression as a baseline, we can graph the distances between points and the line to create a residual plot.
  - Why must a good residual plot look like a random scatter of data points?
    - Good residual plots must treat the differences between observed and predicted as random phenomenon, not a built-in bias. If we see a residual plot with a pattern, we know that the linear model does not adequately capture the distribution.
- Points of Emphasis
  - Residuals are standard deviations from the regression line: It describes how spread out the data is around the linear regression line
  - Since error is minimized with the regression line, the sum of the residuals is 0
    - Method 1: Think about the linear regression line as adjusting the slope and intercept so that a theoretical line creates the smallest total distance from the data points. This line cuts right in the middle of the data so the residual sum is 0.
    - Method 2: To simplify the model, think about the linear regression line as having one point on each side of the line that cancel each other out. In most cases it is the total sum that adds to 0, instead of point to point relationships.

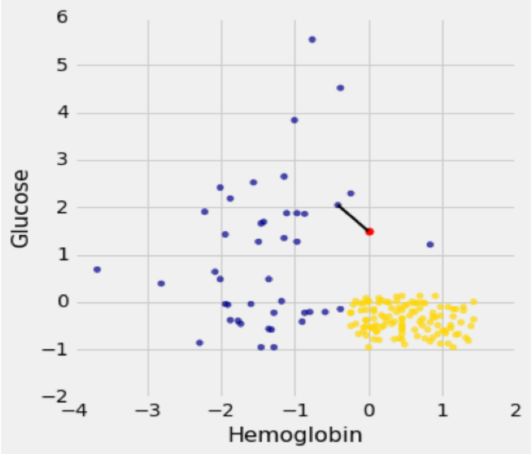
3.9 Linear Regression Inference

- Explanation of Problem
    - Linear regression inference consists of pairs of points, so this is a departure from what we've been learning about bootstrapping beforehand.
    - May exist a disconnect between topics covered before and regression inference even though the process isn't new.
- Pain Points, Problems, and Confusion
    - Why do we sample rows, not just data points randomly?
        * The data are now represented with 2 points. We are interested in the relationship between the x and the y, so it's important to recognize that when we resample, we need to keep them together. Using the sample method on a table will ensure this.
    - Why do we pick the slope, not the intercept?
        * The regression line is made up of the slope and the intercept, so what do we use as the "statistic" that we're generating from each resample? We're gonna choose to use the slope because that tells us something about the correlation between x and y.
    - How do we set up the hypothesis and make conclusions?
        * We want to determine if the slope is 0. This will tell us if we make a conclusion based upon a generated confidence interval. If the confidence interval includes 0, it spans both negative and positive slopes, telling us that it is possible the true slope is 0, or that we can't make a conclusion since we don't know if it is negatively or positively correlated.
        * Null: slope of the true line is 0, alternative: slope of the true line is not 0.
        * We construct a 95% confidence interval for the slope of the regression line, we don't know the true line, so we use bounds for where it could be.
        * If 0 is contained, we fail to reject the null, if 0 is not contained, we can reject the null. If rejected, it is not a good idea to calculate predictions for y based on the x.
- Points of Emphasis
    - Data consists of (x, y) pairs in a table of 2 columns, with some real/true/fixed relationship between the two variables
    - We can describe that relationship more precisely with a line, while we don't know the true line
    - We see a scatter plot with points scattered around the true line pushed off by random noise.
    - Based upon scatter plot we draw the regression line and use as the estimate for the regression line.
    - Code and process-wise, this isn't too different from bootstrapping. The flow of the function should be almost identical. Instead of calculating a test statistic like the mean, we calculate the slope.
- Visuals

3.10 Classification

- Explanation of Problem
    - Students may be confused on how KNN works and why we pick the nearest x number of neighbors
    - Students may have trouble translating their understanding to code.
    - Students may not understand why having too small k values or too big k values could be problematic.
    - Students may not know which k values to pick

- Problems, Pain Points, and Confusion
  - Students may get intimidated if terms like machine learning, classification, etc are thrown around so stress that we are essentially just finding the distance between new points and all given training points and choosing the closest ones.
- Points of Emphasis
  - There are many ways to identity patterns in data, but in this course we use the nearest neighbor method of classification. Conceptually, when we want to classify a new point we find the nearest points to it and classify it with the same attributes.
  - Too small values could be problematic since the classification depends on a single point, which could be affected by random noise or bias, not the culmination of multiple points.
  - Too big values could end up classifying a point on the same number of points, rendering less effective classifications. In extreme situations, it could end up classifying the same result every time
  - Students should pick a k value that balances between the small and big, Usually values 3-7 are good in this class.
  - When teaching students this material make it as clear as possible that we are finding the distance between the new point and the other points. If we are doing nearest neighbor we classify by the nearest point, if it is k-nearest neighbor we use k points.
  - Nearest neighbors is classifying by the closest points in a training set (which is a set where we know for a fact that they are classified correctly).
  - You don't want to train your data on the test set. Ideally, you will only use the test set once.
- Visuals
- in the example below blue points represent individuals "without CKD" and the gold points represent those "with CKD". When adding a new red point, we find that the nearest point is blue. Thus, we classify that individual as "without CKD".

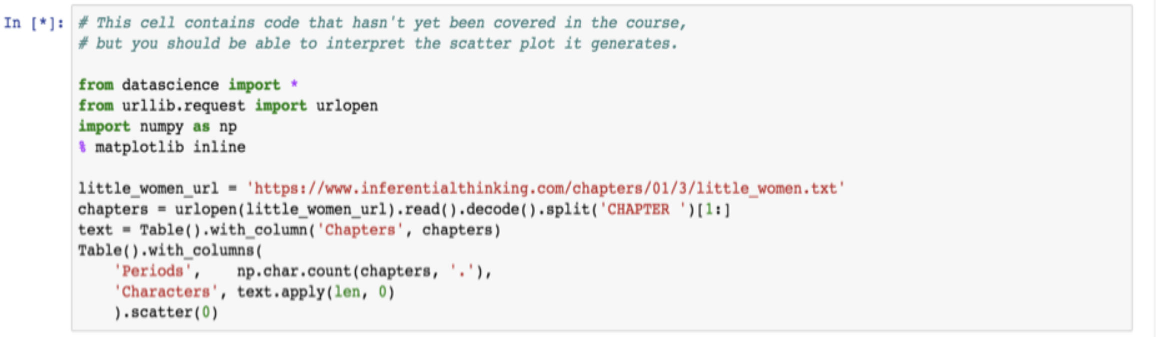


4 Jupyter Troubleshooting

4.1 Technical Problems

- To the left of each cell there is the notation "In [*]" representing the input the and the nth number that has run in this notebook. If it is the first cell being ran once it finished running it should say "In[1]". However, sometimes the cell will keep running, possible forever. In that case it will be represented as

"In[*]".



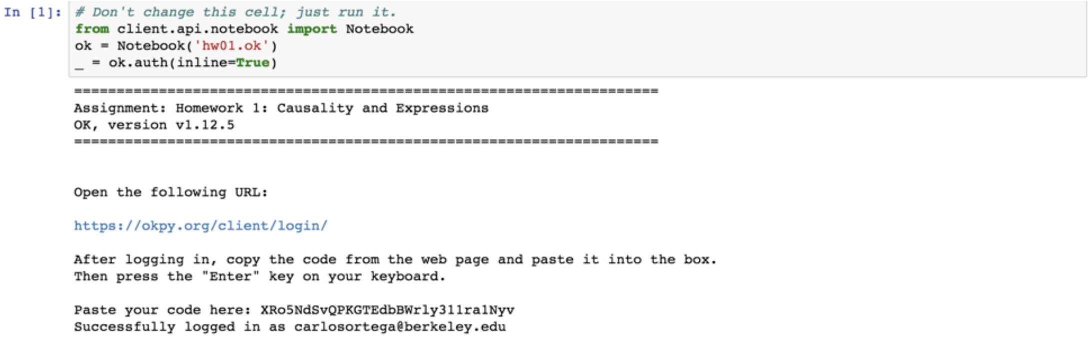
```
In [*]: # This cell contains code that hasn't yet been covered in the course,
        # but you should be able to interpret the scatter plot it generates.

        from datascience import *
        from urllib.request import urlopen
        import numpy as np
        % matplotlib inline

        little_women_url = 'https://www.inferentialthinking.com/chapters/01/3/little_women.txt'
        chapters = urlopen(little_women_url).read().decode().split('CHAPTER ')[1:]
        text = Table().with_column('Chapters', chapters)
        Table().with_columns(
            'Periods',    np.char.count(chapters, '.'),
            'Characters', text.apply(len, 0)
            ).scatter(0)
```

- Troubleshoot in the following order:

  1. Did you run the first cell with okpy? If that cell also has "In[*]" then the cell is still running. Often what has ocurred is that the student has not logged into okpy and copy/pasted the code given to them into the input box that will open below the URL.

     ```
     In [1]: # Don't change this cell; just run it.
             from client.api.notebook import Notebook
             ok = Notebook('hw01.ok')
             _ = ok.auth(inline=True)

             ======================================================================
             Assignment: Homework 1: Causality and Expressions
             OK, version v1.12.5
             ======================================================================


             Open the following URL:

             https://okpy.org/client/login/

             After logging in, copy the code from the web page and paste it into the box.
             Then press the "Enter" key on your keyboard.

             Paste your code here: XRo5NdSvQPKGTEdbBWrly311ralNyv
             Successfully logged in as carlosortega@berkeley.edu
     ```

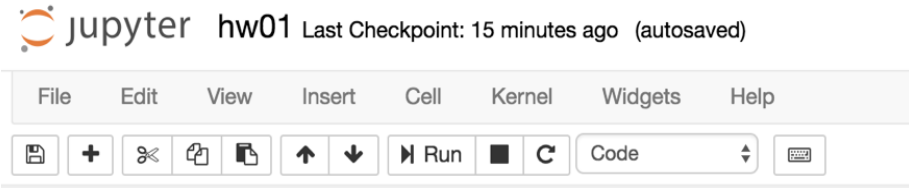
  2. Otherwise let it run for a few minutes (1-2). Sometimes the data is too big.

  3. If it still is running, try shutting down the kernel

     (a) First click Kernel -¿ Interrupt Kernel to try to stop the running code.

     (b) If that doesn't work, click on the Square to attempt to stop the kernel.

     (c) The last trick is to click Kernel -¿ Restart. Try to save the progress before doing this because it does risk losing all work done between now and the last save.

4.2 Debugging

- Ways to debug:

  – Print your variables as you run a function or cell and check if it is something similar to what you expected

  – Check what type of variable you are inputting. Are you inputting a table where a column array should be? `type(x)` will return the data type of x (e.g. table, int, float, string, etc)

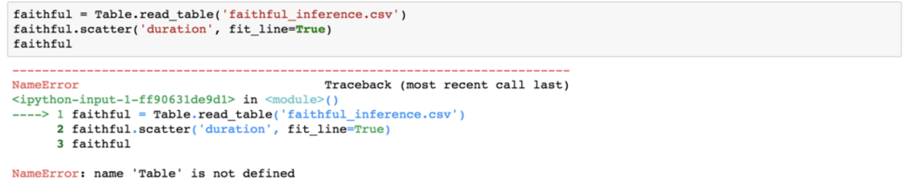
- Common Errors:

  – `TypeError: 'int' object is not callable` or `TypeError: ‘float’ object is not callable`

    - You are attempt to use an int, float, or some other data type as a function. You most likely accidentally redefined a variable that was once assigned to a function to another data type. This can look like:

      ```
      def add_2(x, y):
          return x + y
      add_2 = 3
      add_2(2, 1)
      >>> TypeError: 'int' object is not callable
      ```

  - `NameError: name 'y' is not defined`
    - in these cases you are calling a variable that does not exist. Check the variable names for any misspellings. If you are working with multiple variables with similar names but different in singular/plural form check if you added the 's' where needed.
  - `SyntaxError: unexpected EOF while parsing`
    - The most common cause is a missing or misplaced parentheses or comma. Click besides a parentheses, if it highlights green that means it is closed, but make sure it closes with the correct parentheses, keeping the orders of operation in mind. If it is red, it is not closed and you are either missing parentheses at the end of a statement, or have a surplus of parentheses at the beginning. Check syntax for def statements, for loops, etc.
  - Reading Errors
    - The following is an example of an error.



```
faithful = Table.read_table('faithful_inference.csv')
faithful.scatter('duration', fit_line=True)
faithful

---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-1-ff90631de9d1> in <module>()
----> 1 faithful = Table.read_table('faithful_inference.csv')
      2 faithful.scatter('duration', fit_line=True)
      3 faithful

NameError: name 'Table' is not defined
```

    - The very first line of the error, in red, is the type of error being thrown out.
    - After that will follow one or more blocks of code, but the first is the most helpful. There, you will see an arrow pointing to the line that threw the error.
    - The very last line will repeat the type of error and describe it. In this case, 'Table' is not defined because we did not run the cell that imports the needed libraries

5 Guide Improvement Notes

This document was finished in Spring 2018 and is a starting point. Many of the ideas are not fully developed and much of the guide needs improvement. It will be important for this guide to be continually updated with new topics, corrections, and new methods of teaching. Below are some areas in need of improvement and some topics that should be added to this guide.

- Focus on consistency across different topics
- Future topics
  - Hypothesis Testing
  - Sampling
  - Histograms, Bar Charts, Scatter plots
  - Central Limit Theorem and Law of Averages

6 Credits

Thanks to Dominic Croce, Carlos Ortega, Ravi Singhal, and Maddy Wu for their contributions to this document.

References

[1] Adhikari, Ani, and John Denero *Computational and Inferential Thinking*. 2015. Web. 16 January 2018 2015.

[2] Goodman, S. ”P-Value Diagram.” *Science News*. Annalen der Physik, 322(10):891–921, 1905.

[3] Knuth: Computers and Typesetting,
`http://www-cs-faculty.stanford.edu/~uno/abcde.html`