CSCI-8

DATA SCIENCE

Acknowledgement

We want to thank the Data Science Division at the University of California, Berkeley, for providing the materials in this workbook. These materials represent countless hours of work of Berkeley professors, tutors, undergraduate and graduate student instructors. Special thanks to the entire team at Berkeley's Data Science Education Program for their assistance in creating our course, *CSCI-8: Foundations of Data Science*, and welcoming us into the Data8 Community.

With Gratitude,

Prof. Jin Chon
Prof. Alice Martinez
Prof. Solomon Russell

Table of Contents

Problem Sets

Problem Set 1: By The Numbers

Welcome to your first Data 8 lab! Answer the following questions to the best of your ability. For questions 1 & 2, go to the El Camino Institution Research Student Outcomes page (ECC homepage > Faculty Staff > Institutional Research > Student Outcomes)

1. From the Student Outcomes page click on the *Degrees and Certificates Dashboard.* In the 2019-20 school year, how many women earned an Associates Degree for Transfer from the Business Division?

2. From the Student Outcomes page click on the *Academic Program Review Dashboard.* What course has the third lowest success rate in the Automotive Technology?

3. Day 1 of a ten-thousand-day war is on a Wednesday. What day does it end?

Problem Set 2: Introduction to Tables

Tables are a fundamental way of representing data sets. A table can be viewed in two ways:

- a sequence of named columns that each describe a single attribute of all entries in a data set, or
- a sequence of rows that each contain all information for each attribute about a single entry in a data set.

1. Ready, Willing and Table

Let's look at an example table called `staff`

Name	Year	Semesters on Staff
Rohan	4	7
Katherine	3	5
Alan	4	4
Gregory	3	4
Logan	3	4
Winifred	3	3
Tam	3	4
Connor	3	3
Parham	3	4
Margaret	2	2

The table has 10 rows, each corresponding to one member of Data 8 Staff. Each row has three attributes, the staff member's name, year, and how many semesters they have been on staff. Using just the information from the `staff` table, do we have enough information to generate the following by hand? If not, what additional information do you need?

(You don't need to worry about how you'd do it in Python.)

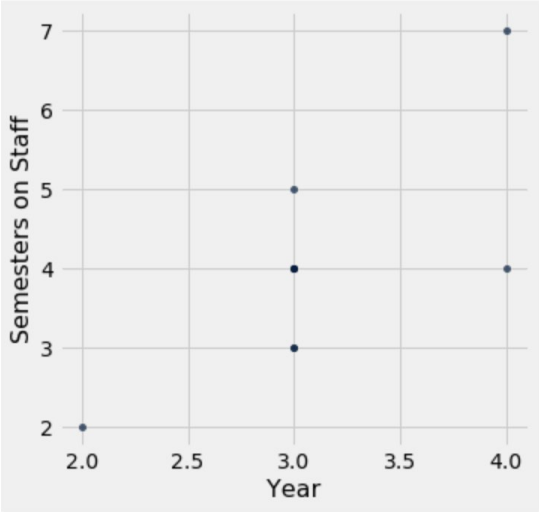
A. True / False

Year	Semesters on Staff average
2	2
3	3.85714
4	5.5

B. True / False

Name	Year
Rohan	Senior
Alan	Senior

C. True / False



2. Casuality, Coworkers and Coffee

Divyesh collected the following information about his coworkers' methods of getting to work and their coffee consumption.

Method	Number of Coworkers	Average Cups of Coffee per Day
Take the Bus to Work	12	1.1
Drive to Work	15	1.9

A. Divyesh is trying to compute the total yearly difference between the cups of coffee that his driving coworkers drink and the cups of coffee his coworkers who bus drink. He will do all of this in a single cell. Identify the errors in the following cell and correct them. *Make sure that the code cell outputs a single positive number.*

```
number_cups_day_difference = (12(1.1 - 15(1.9)))
number_cups_week_difference = number_cups_difference * 7
yearly cups = number_cups_week_difference * 52
```

B. Is there a relationship between transportation method and coffee consumption—an association, a causal relationship or something else? Why?

Problem Set 3: Data Types and Table Manipulation

In lecture, you have been introduced to various *data types* in Python such as integers, strings, and arrays. These data types are particularly important for manipulating and extracting useful information out of data, an important skill for data science. In this section, we'll be analyzing some of the behavior that Python displays when dealing with particular data types.

Discuss each of the following questions with the people around you.

1. What Would Python Do?
For each of the following examples, presume that the code was run in an empty cell. Write down what Python would output. If the code results in an Error, explain why an error would occur.

a. `"I love " + "Data 8"`

b. `"I love Data " + 8`

c. `np.arange(1, 4) + np.arange(2, 7, 2)`

d. `make_array(3, 4, 5) + np.arange(2, 7, 1)`

2. Fun with Arrays
Suppose we have executed the following lines of code. Answer each question with the appropriate output associated with each line of code.

```
odd_array = make_array(1, 3, 5, 7)
even_array = np.arange(2, 10, 2)
```

a. `odd_array + even_array`

b. `odd_array.item(1)`

c. `even_array.item(3) * odd_array.item(1)`

d. `odd_array*3`

In this section, we will practice working with tables. In particular, we will take a look at how to use Python and functions to manipulate tables for them to show what you are interested in.

3. eBay Auctions

Your friend Al is curious to see whether or not it's cheaper to buy his favorite items on eBay rather than through some other platform! Al stumbles upon some auction data from eBay, and decides to use his newly developed Table skills to do some data-crunching. However, Al is making a few mistakes and needs your help. For the following questions, identify why the code won't work as is.

The table below is called `ebay` and contains more than just the 3 rows displayed.

Auction_ID	Item	Opening_Bid	Closing_Price
1	Jacket	50	75
2	Smartwatch	100	150
3	Tablet	350	600

a)
```
# Use comments to describe the code you write
ebay.where('Opening_Bid', are.above(60)) / ebay.num_rows
```

b)
```
ebay.column('Closing_Price') - ebay.select('Opening_Bid')
```

c) Al really wants a new jacket, but his budget is only $150. To see whether eBay has had good deals on jackets historically, Al tries to filter the auction data such that it only contains jackets that were sold for a closing price less than $150. He writes the following code to do so, but hasn't realized the mistake he's making. Help Al fix it so that your friend can hopefully get the jacket he deserves!

```
only_jackets = ebay.where('Item', 'Jacket')
jackets_under_price = ebay.where('Closing_Price', are.below(150))
```

4. Violence in California

You are working on a project related to arrest-related violence in California. After a bit of searching, you finally find some relevant data and import it into your Jupyter Notebook. The table `arrests` is shown below. (Dataset source: https://www.kaggle.com/sohier/arrest-related-violence-in-california)

	Incident_Date_Str	City	Zip_Code	Num_Involved_Civilians	Num_Involved_Officers
0	7/3/2016	Hayward	94544	1	2
1	11/20/2016	San Leandro	94578	1	1
2	5/30/2016	Dublin	94568	1	2
3	6/30/2016	Dublin	94568	1	2
4	5/25/2016	Hayward	94545	1	1

a) Before you start your exploration, you want to understand your data better. For each of the columns **City**, **Num_Involved_Civilians** and **Zip_Code**, identify if the data contained in that column is numerical or categorical.

b) Suppose you were only interested in arrests that involved two, three or four civilians. Assign `range_civilians` to a table that only contains rows from the `arrests` table that correspond to arrests involving two, three or four civilians.

`range_civilians` = _____

c) You're curious about finding out when the arrest involving the most officers occurred in Hayward. Assign `arrest_date` to a string that represents the date of the arrest that occurred in Hayward that involved the most officers. For this question, assume that the maximum number of officers involved in an arrest in Hayward is unique. It is okay for your code to wrap along the next line for `hay_arrests_sorted`.

`hay_arrests_sorted` = _____

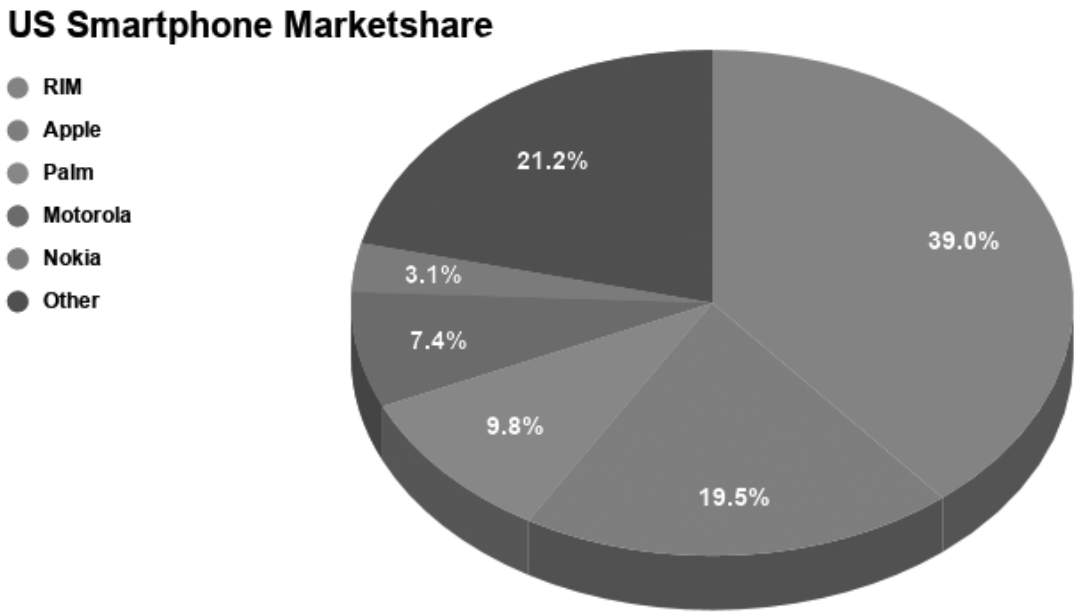
_____

`arrest_date` = _____

Problem Set 4: Visualizations and Histograms

An extremely important aspect of data science is *visualizing* the data in a precise, consistent manner. This week, we will first examine some instances of bad visualizations, and think about how we can improve them. Then, we will transition to focus on *histograms*, which are powerful visualizations used to display the distribution of values for numerical data.

Question 1. The following graphic is a recreation of a graphic presented by Steve Jobs in a keynote at Macworld in 2008. Discuss the graph below with your neighbors, then answer the questions below. (Source: https://www.wired.com/2008/02/macworlds-iphon/)



a) What features could potentially make this visualization misleading?

b) Suppose the underlying data was accessible to you. How would you choose to visualize the data?

Question 2. The table below shows the distribution of rents paid by students in a college town. The first column consists of ranges of monthly rent, in dollars. Ranges include the left endpoint but not the right. The second column shows the percentage of students who pay rent in each of the ranges.

Dollars	Student (%)
250-350	25
350-550	25
550-950	25
950-1350	25

a) Draw a histogram of the data. You do not have to be precise with your drawing, but try your best! Make sure you label your axes!

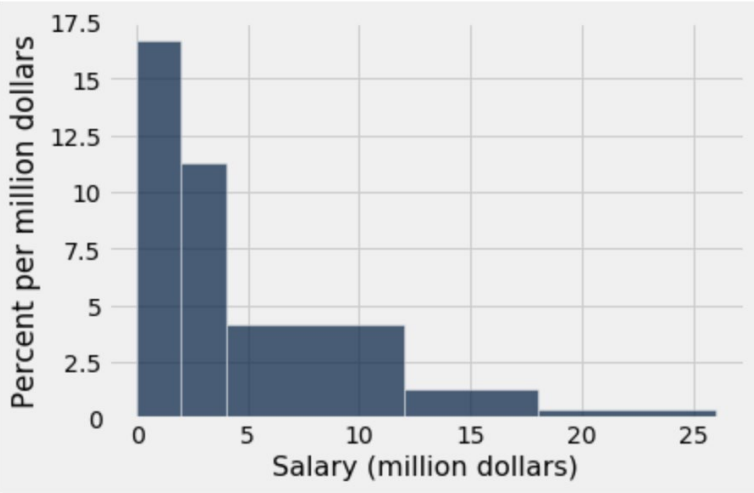
b) What is the height of the bar over the bin 350-550 on the density scale, in the correct units?

A. 12.5% per student
B. 0.125% per student
C. 0.125% per dollar
D. 12.5% per dollar

c) True or false (explain): The data show that the rents are evenly distributed over the interval 250-1350.

d) True or False (explain): The data show that the rents are evenly distributed over the interval 550-950.

Question 3. The table `nba` has a column labeled `salary` containing the 2015-2016 salaries of NBA players. The following histogram was generated by calling `nba.hist(...)`. Also included below is a table with the bins and their corresponding heights.



Bin (million dollars)	[0,2)	[2,4)	[4,12)	[12,18)	[18, 26)
Height (percent per million dollars)	17.49	11.39	3.60	1.60	0.45

The interval `[a,b)` contains all values that are greater than or equal to a and less than b.

a) Which range contains more players: `[0,4)` or `[4,18)`? How many players are in this range? Explain your choice.

Problem Set 5: Iteration and Conditionals

Welcome to Lab 5! This week we will be discussing conditional statements and iteration, which are powerful computational tools that we will use throughout the course. Conditional statements allow data scientists to make more complex decisions with their code, while for loops allow us to repeat the same action many times.

Question 1. What does the following function do? Fill out the docstring description for the function (the first line). *Hint: try to figure out what the function would do on different inputs.*

```
def mystery_function(n1, n2):
      '''_____'''
      if n2 - n1 > 0:
            return n2 - n1
      elif n2 - n1 < 0:
            return n1 - n2
      else:
            return 0
```

Question 2. The instructor of a lower division statistics class has assigned you a task: make a function that takes in a student's score on a scale from 0 to 100 and assigns a letter grade based on the following grade boundaries.

Score	Letter Grade
0 - 69	F
70- 79	C
80 - 89	B
90 - 100+	A

Complete the function `compute_letter_grades`. It takes in a student's score and returns the letter grade they should receive.

```
def compute_letter_grades(score):
      '''
      compute_letter_grades(10)
      >>> "F"
      compute_letter_grades(99)
      >>> "A"
      '''
      if _____________________________:
            return _____________________________
      elif _____________________________:
            return _____________________________
      elif _____________________________:
            return _____________________________
      else:
            return _____________________________
```

Question 3. Skeleton code for the function `count_evens` is below. The function takes in an array of numbers and returns the number of even numbers in the array.

a. Use a combination of iteration and conditionals to complete the function below.

Hint: the % operator returns the remainder if you divide by a certain number! Example: 11 % 5 = 1

```
def count_evens(n_array):
    num_evens = _____________________________

    for ____________________________________________:
       if ____________________________________________:
           ____________________________________________

    return ____________________________________________
```

b. Use array operations to complete the function below.

```
def count_evens(n_array):
    remainder_array = ____________________________________________
    return ____________________________________________
```

Question 4. Complete the function `separate_numbers`, which takes in an array of numbers and a boolean value. It should return the number of even values in the array if the argument `return_even` is True, or the number of odd values in the array if `return_even` is False.

Hint: Use the `count_evens` function you defined above!

```
def separate_numbers(n_array, return_even):
    num_evens = ____________________________________________
    if ____________________________________:
        return ____________________________________________
    else:
        return ____________________________________________
```

Problem Set 6: Hypothesis Testing

When we observe something different from what we expect in real life (i.e. four 3's in six rolls of a fair die), a natural question to ask is "Was this observed difference from what we expect due to random chance? Or was it due to something other than random chance?"

Hypothesis testing allows us to answer that question in a scientific and consistent manner, using the power of computation and statistics to conduct simulations and draw conclusions from our data.

Question 1. Francie is flipping a coin. She thinks it is fair, but is not sure. She flips it 10 times, and gets heads 9 times.

She wishes to determine whether the coin was actually unfair, or whether the coin was fair and her result of 9 heads in 10 flips was by random chance.

1. What is a possible model that you can simulate under?

2. What is an alternative model for Francie's coin? You don't necessarily have to be able to simulate under this model.

3. What is a good statistic that you could simulate? Calculate that statistic for your observed data.
   Hint: If the coin was unfair, it could be biased towards heads or biased towards tails.

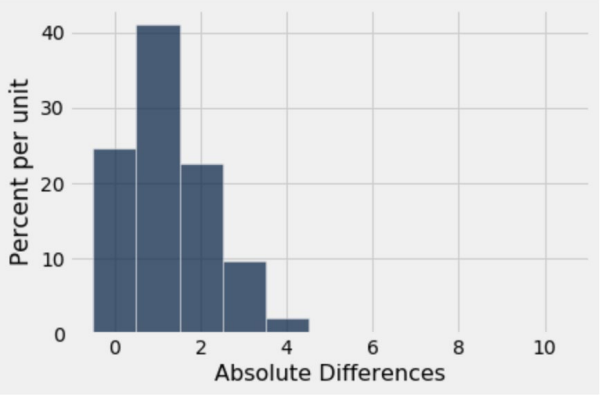
4. Complete the function `flip_coin_10_times`, which takes no arguments and returns the absolute difference between the number of heads in 10 flips of a fair coin and the expected number of heads in 10 flips of a fair coin.

```
def flip_coin_10_times():
   probabilities = make_array(0.5, 0.5)
   proportions = sample_proportions(_____________________________)
   num_heads = ___________________________________________________
   return ________________________________________________________
```

5. Complete the code below to simulate the experiment 10000 times and record the statistic in each of those trials in an array called `abs_differences`.

```
trials = ____________________________________________________________
abs_differences = ____________________________________________________

for ____________________________________:
    abs_diff_one_trial = ____________________________________________
    abs_differences = _______________________________________________
```

6. Suppose we performed the simulation and plotted a histogram of `abs_differences`. The histogram is shown below.



Is the observed statistic described in the question consistent with the model we simulated under?

Question 2. As a student fed up with waiting times at office hours, you scout out the number of people in office hours (OH) from 11-12, 12-1, and 1-2 in B6 Evans. The Head GSI claims that the distribution of students is even across the three times, but you do not believe so. You observe the following data:

OH Time	Number of Students
11-12	250
12-1	300
1-2	200

Being a cunning Data 8 student, you would like to test the Head GSI's claim. Before you design your test, consider: <u>are office hour times numerical data or categorical data?</u>

a. What is the Head GSI's hypothesis?

b. What is the student's hypothesis?

c. Which hypothesis (Head GSI or student) can you simulate under?

d. What is a good statistic to use? *Hint: What is a good statistic for measuring the distance between two categorical distributions?*

Problem Set 7: A/B Testing

One special kind of hypothesis test we do in this class is called an A/B test. The steps used to run an A/B test are the same as a general hypothesis test, but A/B tests have a specific null hypothesis (that two samples were drawn from the same distribution), which we test by performing a *permutation*.

1. Choose True/False for each of the statements below, and explain your answer.

    a) A/B testing is used to determine whether or not we believe two samples come from the same underlying distribution.

    b) To conduct a permutation test, you should sample your data with replacement with a sample size equal to the number of rows in the table.

    c) A/B testing is the same as using total variation distance as a test statistic for a hypothesis test.

    d) A/B testing is about comparing two samples, whereas regular hypothesis testing is about testing properties of one sample.

2. Natalia, a museum curator, has recently been brought specimens of caddisflies collected from various parts of Northern California. The scientists who collected the caddisflies think that caddisflies collected at higher altitudes tend to be bigger. They tell her that the average length of the 560 caddisflies collected at high elevation is 14mm, while the average length of the 450 caddisflies collected slightly lower down is 12mm. She's not sure that this difference really matters, and thinks that this could just be the result of chance in sampling.

    a) How could you test the null hypothesis in the A/B test from above? What assumption would you make to test the hypothesis, and how would you simulate under that assumption?

    b) What would be a useful test statistic for the A/B test?

    c) Assume `original_table` refers to the following table:

A/B label	Specimen length (mm)
High elevation	12.3
Low elevation	13.1
High elevation	12.0

...
(1007 rows omitted)

Fill in the blanks in this code to generate one value of the test statistic under the null hypothesis.

```
def permutation_test():
      shuffled_labels =
      original_table._____________(_________________).column('A/B label')

      original_with_shuffled_labels = original_table.drop('A/B
      label').with_columns(_____________, _________________)

      grouped =
      original_with_shuffled_labels._________(___________,___________)

      means = grouped.column('Specimen length mean')
      statistic = ____________________________________________________
      return statistic
```

d) Fill in the code below to simulate 10000 trials of our permutation test

```
test_stats = ____________________________________________
repetitions = ___________________________________________

for i in np.arange(___________):

      one_stat = ____________________________________

      test_stats = ____________________________________

test_stats
```

e) Given that the p-value of the test was 0.1, draw a possible histogram of the test statistics

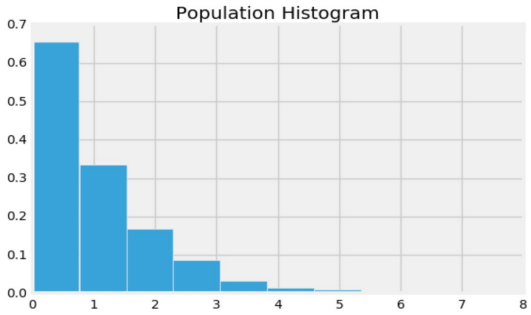
Problem Set 8: Sample Means and Correlation

So far in the course, you have studied multiple different statistics that you can calculate from a sample, including the maximum, median, and the mean. You are now capable of building *empirical distributions* of these different statistics. However, calculating the empirical distribution of the *sample mean* is unique. If you draw a large random sample **with replacement** from a population, then, regardless of the distribution of the population, the probability distribution of the sample mean is roughly normal, centered at the population mean.

Furthermore, the *standard deviation* (spread) of the distribution of sample means is governed by a simple equation, shown below:

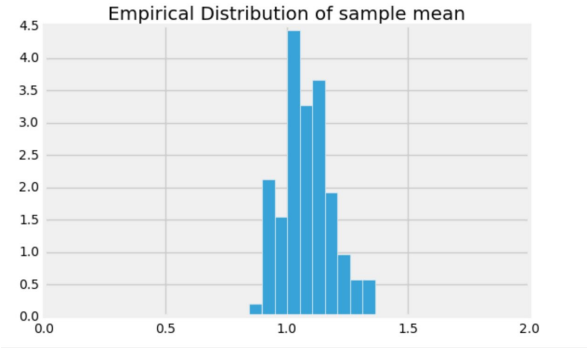
$$\text{SD of all possible sample means} = \frac{\text{Population SD}}{\sqrt{\text{sample size}}}$$

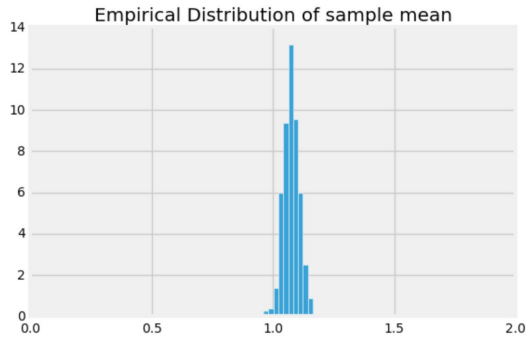
Question 1. Assume that you have a certain population of interest whose histogram is at right.



a) Caroline takes multiple random samples with replacement from the population with the goal of generating an empirical distribution of the sample mean. What shape do you expect this distribution to have? Which value will it be centered around?

b) Suppose that Caroline creates two empirical distributions of sample means, with different sample sizes. Which distribution corresponds to a larger sample size? Why?





c) Suppose you were told that the distribution on the left has a standard deviation of 0.03 and was generated based on a sample size of 100. How big of a sample size would you need if you wanted the standard deviation of my distribution of sample means to be 0.003 instead?

Question 2. You are working with Colby on constructing a confidence interval for the mean height of all Berkeley students. Colby tells you that the empirical distribution of the mean height generated through bootstrapping a sample of size 200 is roughly normal with **mean 170 cm** and **SD 10 cm**. Use this information to construct an approximate 95% confidence interval.

Hint: If you know the empirical distribution is roughly normal, what do you know about the proportion of values that lie within a few SDs of its mean?

Correlation

An important aspect of data science is using data to make *predictions* about the future, using information that we currently possess. A question one might ask would be "Given the US GDP of every year of the previous decade, how can we predict the US GDP for next year?" In order to answer this question, we will investigate a method of using one variable to predict another by looking at the *correlation* between two variables.

Question 3. Why do we convert data to standard units?

Question 4. Write a function called `convert_su` which takes in an array of elements called `data` and returns an array of the values represented in standard units.

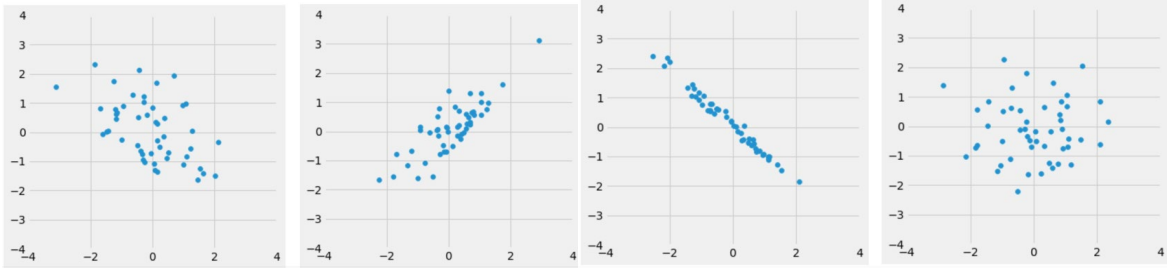
```
def convert_su(data):
```

Question 5. Now let's write a function called `correlation_coefficient` that takes in two arrays `x` and `y` of the same length, and returns the correlation coefficient between the two. *Hint: Feel free to use the function you wrote in the previous question.*

```
def correlation_coefficient(x, y):
```

Question 6. Look at the following four datasets. Rank them from least correlated to most correlated *in magnitude*.

A B C D



Problem Set 9: Linear Regression

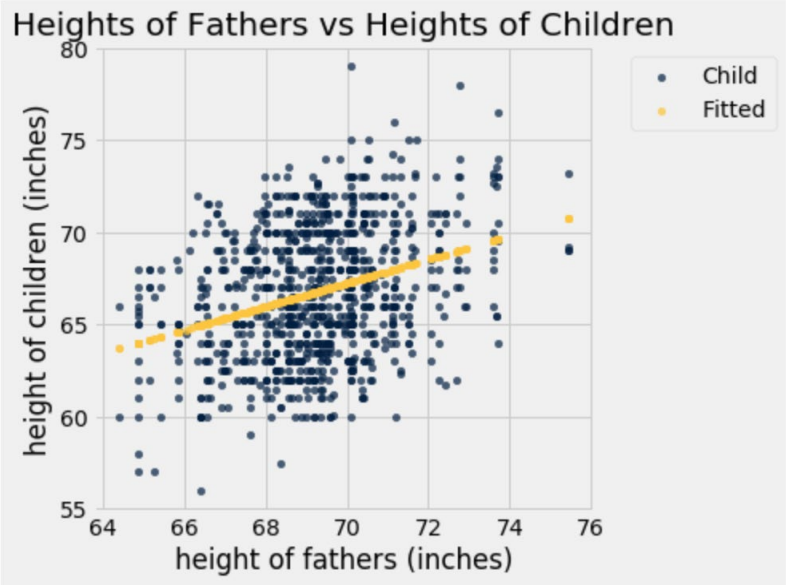
In the previous worksheet, we introduced correlation as a way of quantifying the strength and direction of a linear relationship between two variables. However, the correlation coefficient can do more than just tell us about how clustered the points in a scatter plot are about a straight line. It can also help us define the straight line about which the points (in original units) are clustered, also known as the *regression line*.

The formula for the *slope* and *intercept* for the regression line are shown below. In fact, by a remarkable fact of mathematics, the line uniquely defined by the slope and intercept below is *always* the best straight line for prediction.

$$\textbf{slope of the regression line} = r \cdot \frac{\text{SD of } y}{\text{SD of } x}$$

$$\textbf{intercept of the regression line} = \text{average of } y - \text{slope} \cdot \text{average of } x$$

Question 1. Suppose you are given the scatter diagram shown below that shows the relationship between the height of fathers and the height of children. You have calculated the line of best fit (shown in yellow). Suppose you encounter a new family where the father has a height of 70 inches. How would you predict the height of the children in that family?



Question 2. We want to investigate the correlation between the daily ounces of coffee consumed by an individual and the number of hours the individual stayed awake on that day. It is our intention to use the ounces of coffee consumed to predict the number of hours the individual stayed awake. The data from our sample of 500 people has the following characteristics:

- The number of ounces of coffee consumed has a mean of 12 ounces and SD of 4
- The number of hours stayed awake has a mean of 16 and an SD of 2
- The correlation between the number of ounces of coffee consumed and number of hours spent awake is 0.5.
- Suppose the scatter plot is roughly linear.

a) What is the slope of the line of best fit?

b) What is the intercept of the line of best fit?

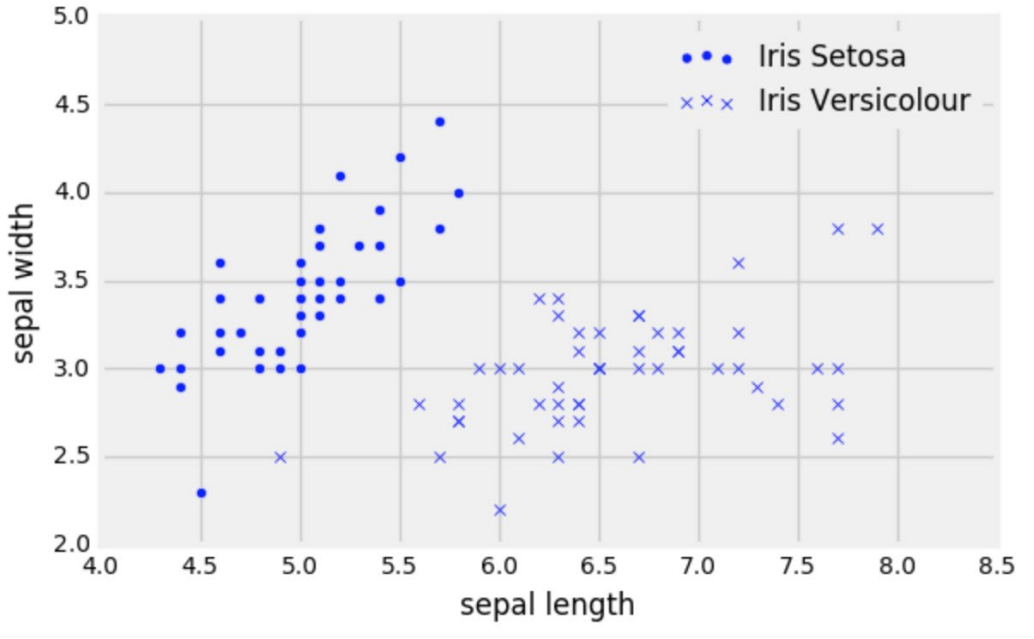
c) Suppose your friend is in this population. She told you that she consumed 24 ounces of coffee that morning. Use your line of best fit to predict how many hours she will stay awake today.

Problem Set 10: Classification, k-Nearest Neighbors and Conditional Probability

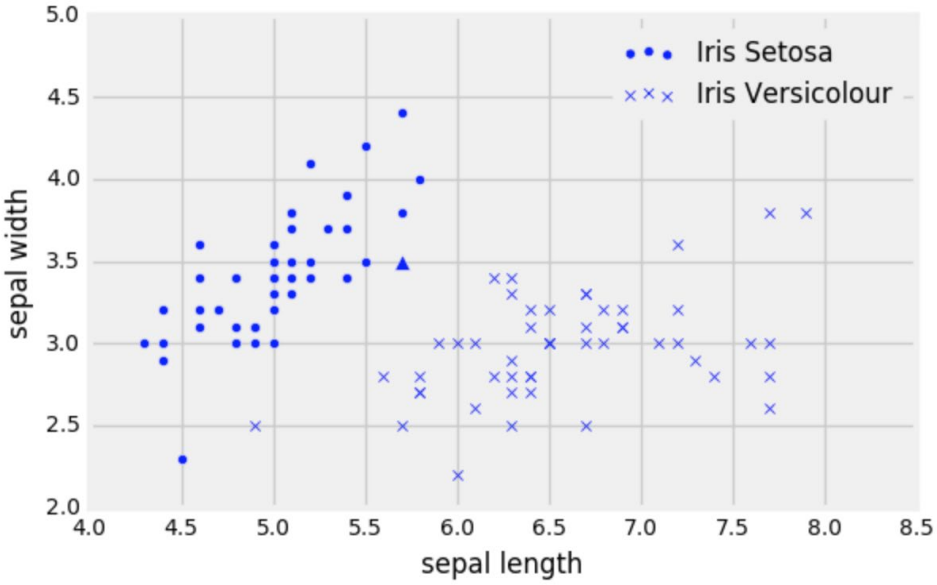
Given the text of an email, how would you determine whether the email is malicious or safe? Perhaps the kinds of words that are used, or the time the email is sent? In this worksheet, we'll discuss *classification*, a term that describes a set of methods and techniques to answer questions like the one above.

Question 1. R. A. Fisher collected a dataset of Iris flowers, which contains two types of iris flowers (Setosa or Versicolor) and the measurements for the sepal width and sepal length. Your goal is to create a classifier that predicts the correct flower type given a new flower.

a. Krista begins by attempting to classify a new flower as an Iris Setosa or an Iris Versicolor based on the sepal length and sepal width of the flower. Draw the decision boundary that the k nearest neighbors algorithm (with k = 3) would generate for this problem.



b. Now Krista wants to classify a new flower (represented as a triangle in the scatter plot on the next page). Describe the steps she would take to classify this new point based on a k nearest neighbors classifier with k=3.



c. Deven suggests that Krista should use a different k for her classifier because he says 3 is too small. What values of k should she avoid?

d. When trying to develop a classifier, we split our original dataset into a training and a test set. We don't look at or use the test set until we have finished training. Why is that a good idea in general? What might happen if we didn't?

e. Suppose Krista chooses k=1 and calculates the accuracy on the training set. Assume that she does **not** remove the point she's trying to classify from the training set when calculating the accuracy. What will the accuracy be on the training set? Will it be representative of the accuracy on the test set?

Question 2. After seeing how successful Krista's K-NN classifier is, Gregory, the owner of an e-commerce store, wants to classify all customers in one of two classes A or B. To do that he will use the following features.

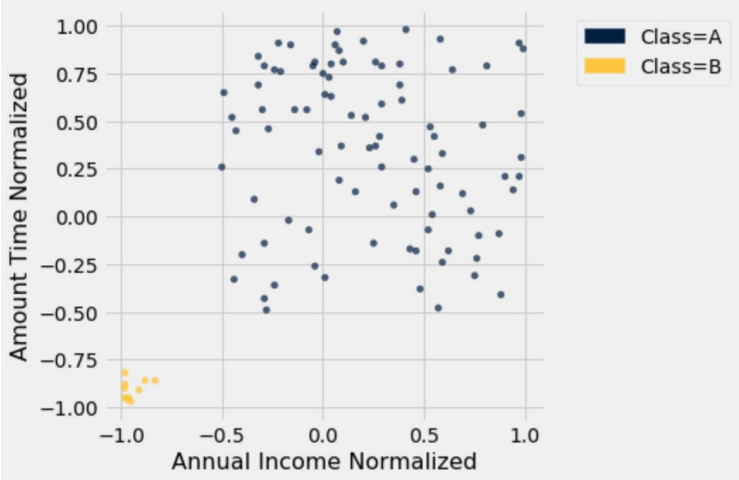
- Annual income of each customer (in dollars)
- The average amount they spend every time they visit his website.
- Their age

a. Gregory wants to run a k nearest neighbors classifier but his friend Roshan claims that he may need to preprocess your data somehow before doing that. What could the problem be and how should he resolve it?

b. Suppose the training set has 100 customers and has the following distribution:

- A: 90% of customers
- B: 10% of customers

We produce the following scatterplot of the training set:



Gregory builds a k-NN classifier for this data with k = 21. What would the accuracy of the classifier be in this scenario?

After implementing his classifier with a different k, Gregory runs the classifier on 1000 customers and finds that:

- 501 of the A customers were classified correctly
- 208 of the B customers were classified correctly
- 104 of the A customers were classified incorrectly
- 187 of the B customers were classified incorrectly

c. Find the following probabilities:

I. Given that a customer was classified incorrectly, the likelihood that they are a B type customer

II. The probability that a customer is an A type customer

III. The probability that a customer is classified correctly

IV. The probability that a customer is classified correctly given that they are an A type customer.

Problem Set 11: Midterm Review

Tables

You are given the following table called `pokemon`. For the following questions, fill in the blanks.

Name	Type	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
Bulbasaur	Grass	318	45	49	49	65	65	45	1	False
Ivysaur	Grass	405	60	62	63	80	80	60	1	False
Venusaur	Grass	525	80	82	83	100	100	80	1	False
VenusaurMega Venusaur	Grass	625	80	100	123	122	120	80	1	False
Charmander	Fire	309	39	52	43	60	50	65	1	False
Charmeleon	Fire	405	58	64	58	80	65	80	1	False
Charizard	Fire	534	78	84	78	109	85	100	1	False
CharizardMega Charizard X	Fire	634	78	130	111	130	85	100	1	False
CharizardMega Charizard Y	Fire	634	78	104	78	159	115	100	1	False
Squirtle	Water	314	44	48	65	50	64	43	1	False

... (790 rows omitted)

1. Find the name of the pokemon of type `Water` that has the highest HP.

```
water_pokemon = pokemon._____(_____,_____)

water_pokemon._____(_____,_____).column("Name").item(0)
```

2. Find the proportion of pokemon of type `Fire` in the dataset whose Speed is strictly less than 100.

```
fire_pokemon = pokemon._____(_____,_____)
fire_pokemon._____(_____,_____)._____/_____
```

3. Return a table containing `Type` and `Generation` that is sorted in decreasing order by the average `HP` for each pair of `Type` and `Generation`.

```
d = pokemon._____(_____,_____)

d.sort("HP mean",_____)._____(_____,_____)
```

4. Find the largest difference of average HP between consecutive generations of Pokemon.

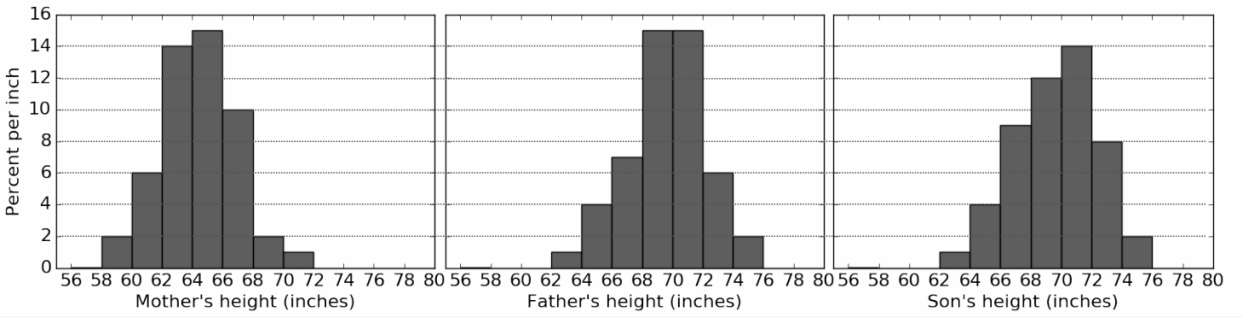
```
generation = pokemon._________(_________________,_____________)\

.sort("Generation",descending=False)

_________(np.diff(_________________._________("HP mean")))
```

5. Return an array that contains ratios of legendary to non-legendary pokemons for each generation.

```
t = pokemon._______________(____________________,____________________)

ratio = t._______________(_________________)/t._______________(_______________)
```

Histograms

Galton measured the heights of the members of **200 families** that each included 1 mother, 1 father, and some varying number of adult sons. The three histograms of heights below depict the distributions for all mothers, fathers, and adult sons. All bars are 2 inches wide. All bar heights are integers. The heights of all people in the data set are included in the histograms.



1. Calculate each quantity described below or write *Unknown* if there is not enough information above to express the quantity as a single number (not a range). Show your work!

a. The **percentage** of mothers that are at least 58 inches but less than 62 inches tall.

b. The **percentage** of fathers that are at least 62 inches but less than 65 inches tall.

c. The **number** of sons that are at least 72 inches tall.

d. The **number** of mothers that are less than 70 inches tall.

2. If the father's histogram were redrawn, replacing the three bins from 68-70, 70-72 and 72-to-74 with one bin from 68-to-74, what would be the height of its bar? If it's impossible to tell, write *Unknown*.

3. The percentage of sons that are taller than all of the fathers is between _____ and _____. Fill in the blanks in the previous sentence with the smallest range that can be determined from the histograms, then explain your answer below.

Probability

1. A fair coin is tossed five times. Two possible sequences of results are HTHTH and HTHHH. Which sequence of results is more likely? Explain your answer and calculate the probability that each sequence appears.

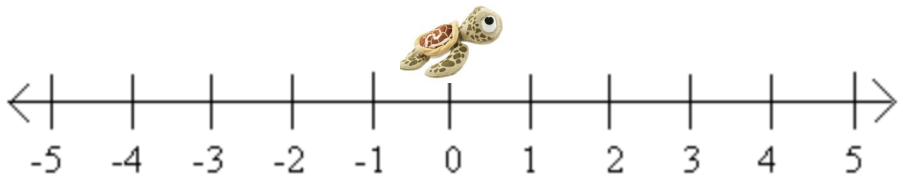
2. Consider a biased coin such that the probability of getting heads is ⅕ and the probability of getting tails is 4/5. The coin is tossed 3 times. What is the probability that you get exactly 2 heads?

3. Once again, we toss the same coin 3 times. What is the probability I get no heads?

4. Again, we toss the same coin 3 times. What is the probability I get at least 1 heads?
Hint: There are two ways of calculating this probability. One is significantly easier to calculate than the other.

Simulation and Hypothesis Testing

Achilles the turtle sits on the number line. Achilles loves long random walks that last a total of 100 time steps. At each time step, Achilles moves based on the following scheme: He flips a coin and moves one step to the right if the coin comes up heads or one step to the left if the coin comes up tails.

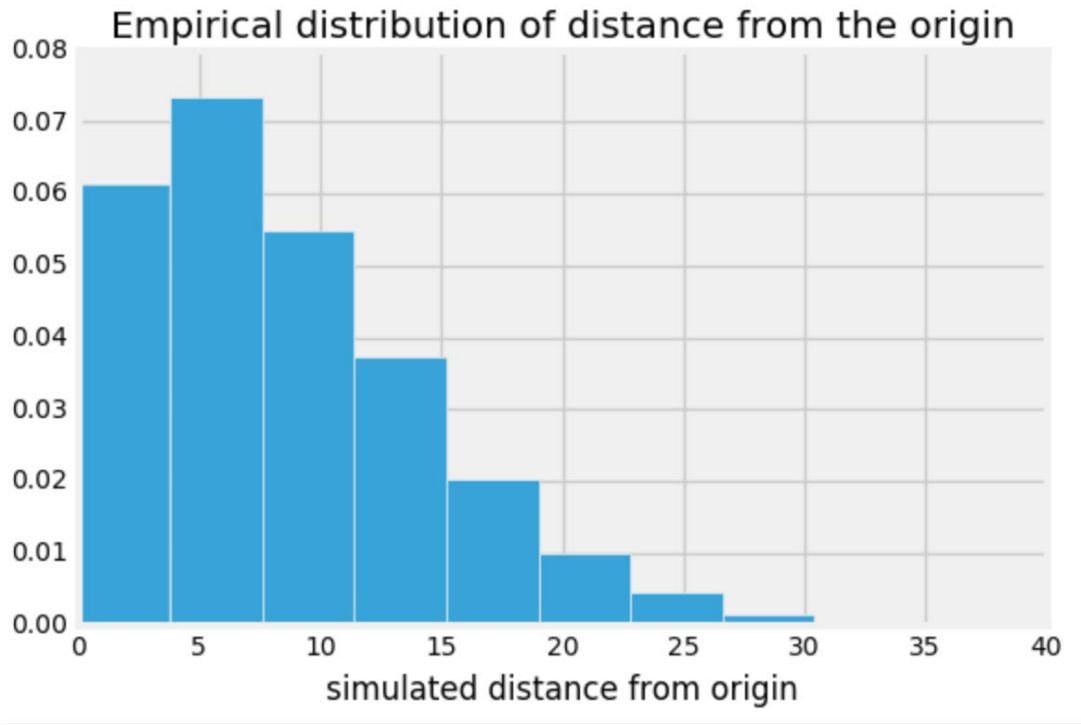


1. Assuming that Achilles' coin is fair, write a function called `one_walk` that simulates one random walk of 100 time steps and returns how far from the origin Achilles ends up at the end of his walk. You may assume that Achilles always starts from the origin.

```
def one_walk():
```

2. Assuming that Achilles' coin is fair, we would like to simulate what would happen if Achilles took 10000 different random walks. Complete the simulation below and keep track of how far Achilles ends up from the origin in each of his walks in an array called `distances`. The histogram shown below is an example of a histogram plotted from `distances`.

```
distances = make_array()

for i in np.arange(10000):
    new_distance = _____________________________________________________
    distances = _____________________________________________________
```



3. Achilles goes for a walk and claims that at the end of his walk, he ended up 30 steps away from the origin. You notice this is strange, so you want to run a hypothesis test to test whether or not Achilles used a fair coin. Fill in the blanks below for the null and alternative hypotheses as well as a good test statistic for this experiment.

Hint: When considering your alternative hypothesis, note that we do not really care about whether the coin is biased towards heads or towards tails.

Null Hypothesis:

Alternative Hypothesis:

Test Statistic:

4. Write the code to calculate the p-value given the test statistic listed above and using a 5% p-value cut-off. Then, describe the different conclusions that you would arrive at depending on the p-value.

Hint: We simulated an array in part(b) of test statistics under the null hypothesis. Try to use the `distances` *array.*

```
p_value = _____________________________________________________________
```

True/False

Respond with true or false to the following questions. If your answer is false, explain why.

1. In the U.S. in 2000, there were 2.4 million deaths from all causes, compared to 1.9 million in 1970, which represents a 25% increase. The data shows that the public's health got worse over the period 1970-2000.

2. A company is interested in knowing whether women are paid less than men in their organization. They share *all* their salary data with you. An A/B test is the best way to examine the hypothesis that all employees in the company are paid equally.

3. Consider a randomized control trial where participants are randomly split into treatment and control groups. There will be no systematic differences between the treatment and control groups if the process is followed correctly.

4. A researcher considers the following scheme for splitting a people into control and treatment groups. People are arranged in a line and for each person, a fair, six-sided die is rolled. If the die comes up to be a 1 or a 2, the person is allocated to the treatment group. If the die comes up to be a 3, 4, 5 or 6 then the person is allocated to the control group. This is a randomized control experiment.

5. You are conducting a hypothesis test to check whether a coin is fair. After you calculate your observed test statistic, you see that its p-value is below the 5% cutoff. At this point, you can claim with certainty that the null hypothesis can not be true.

6. You roll a fair die a large number of times. While you are doing that, you observe the frequencies with which each face appears and you make the following statement: As I increase the number of times I roll the die, the probability histogram of the observed frequencies converges to the empirical histogram.

Bonus Question: What is the Law of Averages? Can you see why it allows us to run large scale simulations instead of trying to find exactly what the probability distribution of a test statistic is?

Multiple Choice

1. Gary is playing with a coin and he wants to test whether his coin is fair. His experiment is to toss the coin 100 times. He chooses the following null hypothesis:

Null Hypothesis: The coin is fair and any deviation observed is due to chance.

For each of the alternative hypotheses listed below, determine whether or not the test statistic is valid.

a. **Alternative Hypothesis:** The coin is biased towards heads.
Test Statistic: # of heads

b. **Alternative Hypothesis:** The coin is not fair.
Test Statistic: # of heads

c. **Alternative Hypothesis:** The coin is not fair.
Test Statistic: |# of heads - expected # of heads|

d. **Alternative Hypothesis:** The coin is biased towards heads.
Test Statistic: |# of heads - expected # of heads|

e. **Alternative Hypothesis:** The coin is not fair.
Test Statistic: ½ - proportion of heads

2. It is now generally accepted that cigarette smoking causes heart disease, lung cancer, and many other diseases. However, in the 1950s, this idea was controversial. The statistician and geneticist R. A. Fisher advanced the “constitutional hypothesis”, which claims there is some genetic factor that predisposes individuals to smoke as well as to die from diseases.

Suppose that Fisher was correct and there is a gene that predisposes individuals towards smoking as well as getting lung cancer. In the context of this experiment, how would you characterize this gene?

A. treatment
B. outcome
C. confounding factor
D. placebo

Fun with Functions

1. Write a function called `compute_pvalue` that given an empirical distribution in the form of an array and the observed value of your test statistic, calculates the p-value for that test statistic. You may assume that large values of your test statistic provide evidence against the null hypothesis.

```
def compute_p_value(empirical_dist, observed_ts):
```

2. Now write a function called `is_significant` that takes in an empirical distribution, the observed test statistic and a p-value cutoff, returns `True` if the p-value of the observed test statistic is statistically significant based on the cutoff provided and `False` otherwise.

Hint: Use the function you defined in Question 1!

```
def is_significant(empirical_dist, observed_ts, cutoff):



    return
```

3. Write a function called `is_prime` that takes in a number `n` and returns `True` if the number is prime and `False` otherwise. Remember that a number is prime if it is only divisible by itself and 1. In general, we do not consider 1 to be a prime number.

Hint: The % operator is your friend.

```
def is_prime(n):



    return
```

More Hypothesis Testing

Chloe is a big fan of Trader Joes' frozen mac n cheese, but she noticed that the cheese used in it varies from box to box. A Trader Joe's employee provides her with some data about the 4 different cheeses used and the probability of them being used in each box:

Cheese	**Probability**
Velveeta	0.05
Gruyère	0.55
Sharp Cheddar	0.25
Monterey Jack	0.15

Chloe is suspicious about this distribution. After all, Velveeta is much cheaper to use than Gruyère, and she has also never bought a box that uses Gruyère. Chloe decides to buy many boxes throughout the next month and tracks the type of cheese used in each box. She uses this to conduct a hypothesis test.

1. Write the correct null hypothesis for this experiment
   - Null Hypothesis:
   - Alternative Hypothesis:

```
observed_proportions = make_array(0.2, 0.3, 0.45, 0.05)
employee_proportions = make_array(0.05, 0.55, 0.25, 0.15)
```

The array `observed_proportions` contains the proportions of cheese that Chloe observed in 20 boxes of Mac n Cheese.

2. Chloe wants to use the mean as a test statistic, but Katherine suggests that she uses the TVD (total variation distance) instead. Which test statistic should Chloe use in this case? Briefly justify your answer. Then write a line of code to assign the observed value of the test statistic to `observed_stat`.

```
observed_stat = _____________________________________________
```

3. Define the function `one_simulated_test_stat` to simulate a random sample according to the null hypothesis and return the test statistic for that sample.

```
def one_simulated_test_stat():
      sample_prop = ____________________________________
      return ___________________________________________
```

4. Chloe simulates the test statistic 10,000 times and stores the results in an array called simulated_stats. The observed value of the test statistic is stored in `observed_stat`. Complete the code below so that it evaluates to the p-value of the test:

```
_____________(simulated_stats _________ observed_statistic) / _____________
```

5. Given that the computed p-value is 0.0825, which of the following are true? Select all that may apply.

a. Using an 8% p-value cutoff, the null hypothesis should be rejected
b. Using a 10% p-value cutoff, the null hypothesis should be rejected.
c. There is an 8.25% chance that the null hypothesis is true
d. There is an 8.25% chance that the alternative hypothesis is true

A/B Testing

Alvin loves the lemon bars served at Crossroads and Cafe 3 dinner. However, they are highly popular among students and often run out. Alvin would like to know if a different number of students go to Cafe 3 or Crossroads (for reasons other than chance), so he can attend the less-populated dining hall and grab a lemon bar without worry.

1. Alvin will use A/B testing to figure out if there is a difference in the dining halls. What null hypothesis, alternative hypothesis and test statistic should Alvin use? Make sure to define how you will simulate the null hypothesis.

Null Hypothesis:

Alternative Hypothesis:

Test Statistic:

2. Alvin collects some data about the distribution of students at Cafe 3 and Crossroads for the semester. He reports the following numbers in the `dining_hall` table (only the first three rows are shown). Write a line of code to find the observed value of the test statistic and assign it to `observered_stat`.

Dining Hall	**Students**
Crossroads	350
Crossroads	420
Cafe 3	280

```
means = _____________________________________________________________
observered_stat = ____________________________________________________________
```

3. Complete the function one_sim_stat to perform one permutation test and return a value of the test statistic.

```
def one_sim_stat():
```

```
    shuffled = ____________________________________________________________
    original_with_shuffled_labels = ____________________________________
    ____________________________________________________________________
    means = ____________________________________________________________
    ____________________________________________________________________
    return _____________________________________________________________
```

4. Generate 10,000 simulated statistics and calculate the empirical p_value.

```
simulated_stats = _____________________
repetitions = 10,000

for ____________________________________________________:
    one_stat = ____________________________________________
    ____________________________________________________

p_value = ____________________________________________________________________
```

5. Describe some potential flaws in the data collection process that Alvin employed. What could bias his results?

Project 1 Problems: Groups, Joins, Pivots

Welcome to Project 1 Lab! This week we will be discussing groups, joins, and pivots. The group function allows us to aggregate the unique entries in one or more columns, while the join function allows us to merge data from two tables into a single table. Alternatively, pivot allows us to aggregate over the unique values of two columns. All of these methods are vital to creating simple yet powerful tables that assist in analyzing data.

Question 1. Ian has opened up a chocolate store where he sells small boxes of chocolates in groups of different sizes and colors. His table `chocolates` is as follows:

Color	Shape	Amount	Price ($)
Red	Round	4	1.30
Green	Rectangular	6	1.20
Blue	Rectangular	12	2.00
Red	Round	7	1.75
Green	Rectangular	9	1.40
Green	Round	2	1.00

Notice that the table contains multiple rows containing information about chocolates of the same color. We would like to figure out how many chocolates of each color he has for sale in total, and what the cost would be to purchase all chocolates of each unique color.

a. Write a line of code that will return a new table which displays the total number of boxes for each color.

b. Write a line of code which will return a new table with the total number of chocolates and the total cost for each unique color. For example, the row for “Red” should have a total of 4+7=11 chocolates, and a total cost of $1.30 + $1.75 = 3.05.

Question 2. The table below, called `weights`, contains information about the weights of the chocolates that are sold. The weights of the chocolates differ depending on the shape, and round chocolates have two different sizes.

Shape	**Weight(g)**
Round	3.1
Round	4.25
Rectangular	3.6
Triangular	2.9

The following line of code has been executed in a blank cell. Take a moment to discuss with your neighbors what the resulting table will look like. Then, write the number of columns and rows in the resulting table, and describe the information in the table in 1-2 sentences.

Hint: It may help to draw a sketch of the resulting table!

```
chocolates.join('Shape', weights)
```

Question 3. We will continue with the same table as before, copied below for your convenience.

Color	Shape	Amount	Price ($)
Red	Round	4	1.30
Green	Rectangular	6	1.20
Blue	Rectangular	12	2.00
Red	Round	7	1.75
Green	Rectangular	9	1.40
Green	Round	2	1.00

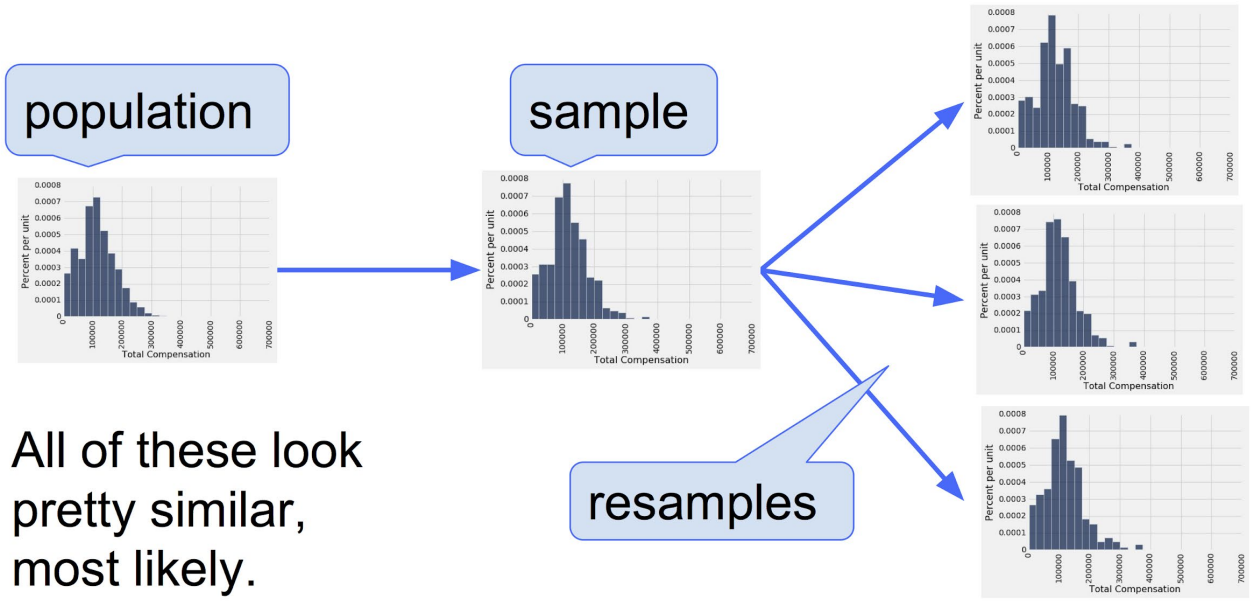
Write code to create a pivot table on the colors and shapes of chocolates, finding the average price for each color-shape combination. Then, fill in the blank table in the image of the resulting table.

Hint: You can use the `np.average` function to find the average of an array of inputs. The average of no values is marked as zero.

	Rectangular	Round
Blue		
Green		
Red		

Project 2 Problems: Bootstrap and Confidence Intervals

Suppose we are trying to estimate a population parameter. Whenever we take a random sample and calculate a statistic to estimate the parameter, we know that the statistic could have come out differently if the sample had come out differently by random chance. We want to understand the *variability* of the statistic in order to better estimate the parameter. However, we don't have the resources to collect multiple random samples. In order to solve this problem, we use a technique called *bootstrapping*.



Question 1: What is the difference between a parameter and a statistic? Which of the two is random?

Question 2: Assume we have one large, random sample. How could we generate another sample that resembles the population if we don't have the resources to sample again from the population?

Tennis

Adith is interested in the height of tennis players. He's collected a sample of 100 heights of professional women's tennis players in the table `tennis`.The average of the sample is 69 inches and the standard deviation is 1.5 inches! He wants to use this sample to make some estimates about the population of the heights of professional women's tennis players.

Question 3: Adith wants to estimate the 50th percentile of heights in the women's professional tennis population. The 50th percentile has another name- the median! Define a function `ci_median` that constructs a 85% confidence interval for the median as follows. The function takes the following arguments:

Hint: to find the median of an array you can use the percentile function or the np.median function.

- tbl: A one-column table consisting of a random sample from the population; you can assume this sample is large
- reps: A number of bootstrap repetitions

```
def ci_median(tbl, reps):
    stats = ________________________________________________
    for ____________________ :
        new_samp = ____________________________________________
        new_median = ___________________________________
        stats = ________________________________________
    left_end = ____________________________________________________
    right_end = ___________________________________________________

    return make_array(left_end, right_end)
```

Question 4: If Adith calls `ci_median(tennis, 2000)` 500 times, approximately how many CI's do we expect to contain the actual median height of women's professional tennis athletes?

Adith and Jessica are arguing about the average height of women's tennis players. Jessica wishes she played tennis professionally, so she argues that the average height of all players is 68 inches. Adith doesn't know if Jessica's guess is too high or low but thinks that Jessica's guess is wrong. Help Adith conduct a hypothesis test about this!

Question 5: Write down the null and alternative hypotheses and test statistic for this test!

Null:

Alternative:

Test Statistic:

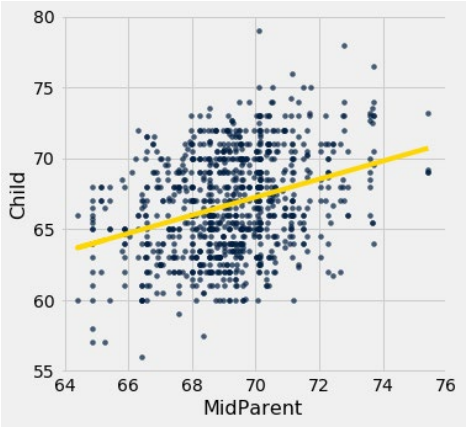
Question 6: Adith doesn't have access to Python and unfortunately can't use a simulation for this hypothesis test. Can he still figure out what the distribution of average heights is? Why? *Hint: What do we know about the population?*

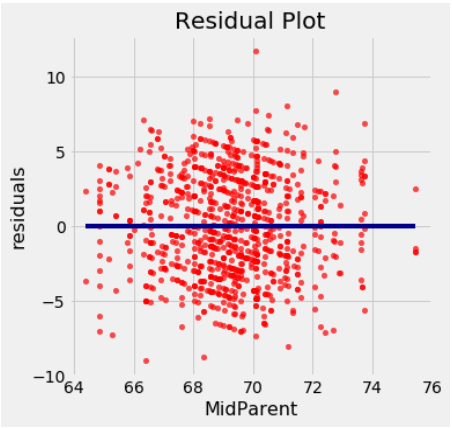
Question 7: Adith decides to do the simulation and the 95% confidence interval he computes is [68.6, 69.2]. What should his conclusion for the hypothesis test be?

Project 3 Problems: Residuals and Regression Inference

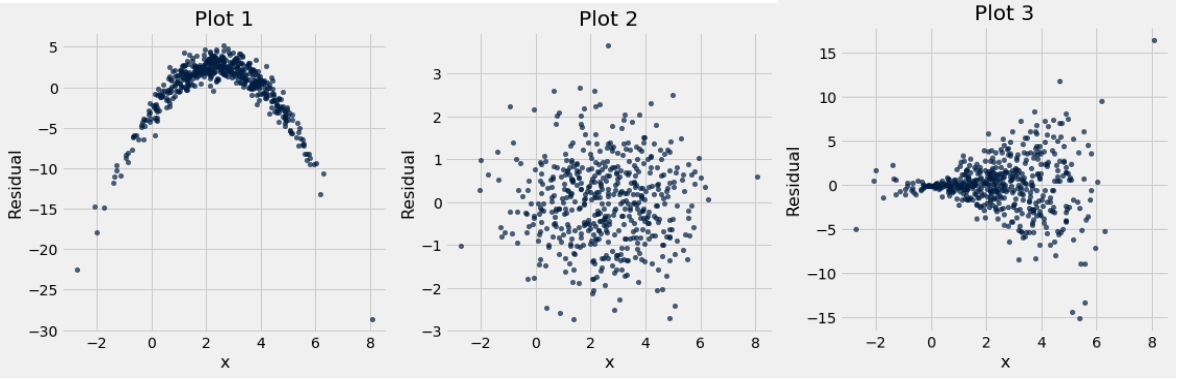
In data science, we can use regression inference in order to make predictions; however, in order to assess the accuracy of our linear regression model, we want to examine the error between our predictions and the actual data. These errors are called *residuals*.

An example can be found below in the graph of midparent heights compared to child heights. The graph of the residuals is shown on the right.





Question 1. Displayed below are three residual plots. For which of the following residual plots is using linear regression a reasonable idea, and why?



Question 2. Yash has a sample of 100 snacks (Yum!). This dataset contains the calories from fat (`cal_fat`) and the calories total (`cal_total`) for each snack. Yash wants to use a snack's `cal_fat` to predict its `cal_total`. The standard deviation of `cal_fat` is 5 calories, and the standard deviation of `cal_total` is 10 calories. The correlation coefficient between the two variables is 0.6.

a. What would be the SD of the residuals between the predicted `cal_total` and the actual `cal_total`?

b. Suppose the correlation coefficient between the two variables was actually 0.9. What would be the SD of the residuals in this case?

c. What does this say about the relationship between the SD of the residuals and the correlation coefficient?

d. Yash thinks that there is no association between `cal_fat` and `cal_total`, and that his sample was just biased. How can Yash test this hypothesis?

Null Hypothesis:

Alternative Hypothesis:

Describe Testing Method:

e. Yash runs his hypothesis test and gets a 99% confidence interval of 0.24 to 089. Should he reject the null hypothesis?

f. Finally, Yash wants to generate a line of best fit for his data. Should he use the method of least squares or the regression equations?

Worksheets

Worksheet 1: Expressions, Data Types, Sequences

1 Expressions and Data Types

Key Concepts

Function calls

Done by writing the name of the function and passing in arguments in the parenthesis. Functions can have any number of arguments.

Example: `min(4,5,6)`

Assignment Statements

Assigns the variable on the left hand side of the equal side to the value of the expression on the right hand side

Example: `var = max(1,3)` first evaluates the right hand side to 3, then assigns it to the variable `var`

Arithmetic Operations

- `+`, `-`, `*`, and `/`
- `%` for remainder
- `**` for exponentiation
- `>`,`<`,`==`,`!=`,`>=`,`<=` for comparison

Data Type Conversion Functions

- `str(...)`
- `int(...)`
- `float(...)`

Practice Problems

1.1 Evaluate the following code snippets

a. `str(8) + str(24)`

b. `abs(1-(4**2))`

c. `min(10%2, 10%3, 10%1)`

d. `int('4')*6`

1.2 Jaylen Brown challenges you to a modified Three-Point Contest. Jaylen shoots 10 shots and his score is equal to the number of baskets he makes. Assume Jaylen makes a number of shots stored in the variable `jaylen_makes`. On the other hand, you get 3 tries to shoot 10 shots each, and your score comes from whichever of the 3 tries has the most shots made. Whoever has the highest score wins.
Assume the results of your attempts are stored in `try1`, `try2`, and `try3`. Assume there are no ties. Write a line of code which returns whether you won the game. (*Hint: this should be* `True` *or* `False`)

1.3 Assume the variable name `eight` has been assigned to the string `'8'`. Using only this string, the string methods, arithmetic, and any type conversion functions (*`int, str, etc.`*), print the square of 88. You may want to use variable assignments so you don't have to reuse code.

1.4 Write a line of code that evaluates whether 111*43 is even.

2 Arrays

Key Concepts

Array

Data type which can hold sequences of data, as long as they all have the same type.
Useful for arithmetic operations, as it allows for mathematical expressions to be applied to all elements to an array at one time.
Useful array methods/functions include:

- *`make_array(...)`*
- `len(array)`
- *`np.arange(start, end, step)`*
- *`arr.item(x)`*
- *`np.cos`*, *`np.log`*, *`np.sin`*, *`np.sqrt`*, `etc.`

Practice Problems

2.1 Using different approaches, write two separate lines of code that evaluate to the first 10 multiples of 3 (starting at 3).

2.2 Assume `shopping` is an array of dollar amounts spent in a store (before tax) by 5 different customers. Write a line of code to answer the following questions.

a. What is the total amount spent by the customers?

b. What was the largest amount spent by a customer?

c. Did person 2 spend more than person 4? Assume there's no person 0. Your code should evaluate to either `True` or `False`.

d. Assume tax is 10 percent on these items. What did each customer spend after tax?

e. What was the absolute difference between Person 1's expenditure before tax, and Person 5's expenditure after tax?

2.3 Use one line of code to figure out what every number from 1 to 10 to the power of 1+ that number is. The first number in your output should be 1^2 = 1, the second number should be 2^3, and the last number should be 10^11.

2.4 Assume we have an array of strings called `str_arr`

a. Find the length of the array.

b. Find the length of the third string in this array.

2.5 Assume we have arrays `first_arr` and `second_arr`, which are arrays of floats. Use one line of code to answer the following questions: How much larger (or smaller) is the sin of the first element in `first_arr` than the cosine of the last item in `second_arr`.

Worksheet 2: Tables and Histograms

1 Tables

Key Concepts

Table Manipulation: Table methods can be applied to manipulate a given table in order to conduct analysis. This is done by writing `tbl.method()` and passing in the necessary arguments. When using table methods, make sure to assign a variable to the result of the new Table method. *Examples*: `where, select, sort, drop, column`

Practice Problems

Assume all imports are completed. **Pay close attention to all of the syntax, as it's difficult to learn at first.** Feel free to ask your tutor any questions you have!

For the first part, we're just going to focus on the `actors` table, which begins like this:

Actor	Total Gross	Number of Movies	Average per Movie	#1 Movie	Gross
Harrison Ford	4871.7	41	118.8	Star Wars: The Force Awakens	936.7
Samuel L. Jackson	4772.8	69	69.2	The Avengers	623.4
Morgan Freeman	4468.3	61	73.3	The Dark Knight	534.9
Tom Hanks	4340.8	44	98.7	Toy Story 3	415
Robert Downey, Jr.	3947.3	53	74.5	The Avengers	623.4
Eddie Murphy	3810.4	38	100.3	Shrek 2	441.2
Tom Cruise	3587.2	36	99.6	War of the Worlds	234.3
Johnny Depp	3368.6	45	74.9	Dead Man's Chest	423.3
Michael Caine	3351.5	58	57.8	The Dark Knight	534.9
Scarlett Johansson	3341.2	37	90.3	The Avengers	623.4

We will start with some simple queries on this table and move our way upwards to more advanced ones.

1.1 Write a line of code that returns `actors` sorted from highest to lowest number of movies.

1.2 Now, write a line of code to find the actor who has made the most movies. Do not return a table with the actor's name; just return the name as a string.

1.3 What is Tom Hanks' #1 movie? Write a line of code to return the name of the movie as a string.

1.4 Write a line of code which returns a table consisting of only the "Actor" column where the elements in the "Actor" column are the names of actors who have above 40 movies and have a total gross below 3000.

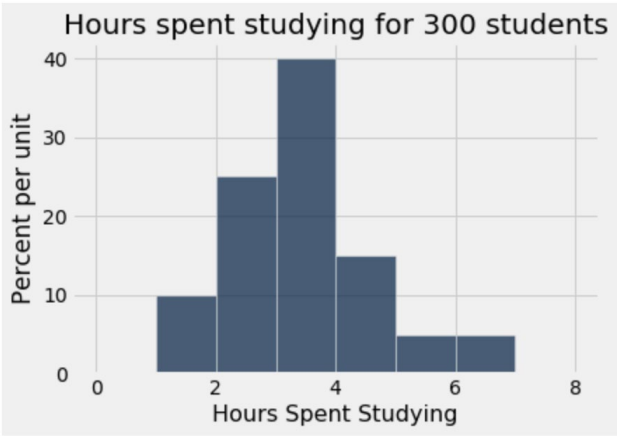
2 Histograms

Key Concepts

Histograms are an important visualization for understanding the *distribution* of a single numerical variable. While they appear similar to bar charts, histograms have important differences which make them powerful visualizations for data science.

Practice Problems

Suppose you are interested in the number of hours, on average, that UC Berkeley students spend studying a day. You survey 300 random UC Berkeley students, record the number of hours studying a day they reported, and plot a histogram with the data. The histogram is shown below.



2.1 What percentage of students studied between two and three hours a day?

2.2 How many students studied between three and four hours a day?

2.3 Suppose you created a new bin for students who studied between three and five hours a day. What would be the height of the new bar?

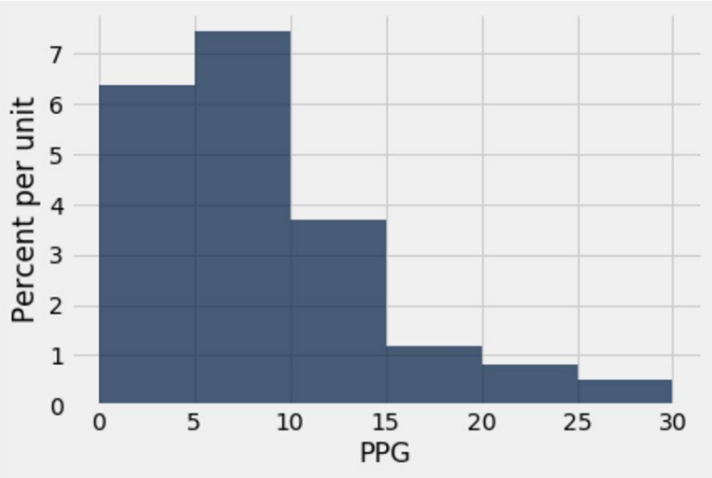
Throughout this section, we'll be focusing on the following table: `nba.csv`. It describes the average statistics of NBA players for the 2016-2017 season. Pay special attention to the syntax! This can quickly become confusing, so ask your tutor if anything seems confusing.

The first few rows of the `nba` table look like this:

Rk	Player	Pos	Age	Tm	G	GS	MP	FG	FGA	FG%	3P	3PA	3P%	2P	2PA	2P%	FT	FTA	FT%	TRB	AST	STL	BLK	TOV	PF	PPG
1	Alex Abrines	SG	23	OKC	68	6	15.5	2	5	0.393	1.4	3.6	0.381	0.6	1.4	0.426	0.6	0.7	0.898	1.3	0.6	0.5	0.1	0.5	1.7	6
2	Quincy Acy	PF	26	TOT	38	1	14.7	1.8	4.5	0.412	1	2.4	0.411	0.9	2.1	0.413	1.2	1.6	0.75	3	0.5	0.4	0.4	0.6	1.8	5.8
3	Steven Adams	C	23	OKC	80	80	29.9	4.7	8.2	0.571	0	0	0	4.7	8.2	0.572	2	3.2	0.611	7.7	1.1	1.1	1	1.8	2.4	11.3
4	Arron Afflalo	SG	31	SAC	61	45	25.9	3	6.9	0.44	1	2.5	0.411	2	4.4	0.457	1.4	1.5	0.892	2	1.3	0.3	0.1	0.7	1.7	8.4
5	Alexis Ajinca	C	28	NOP	39	15	15	2.3	4.6	0.5	0	0.1	0	2.3	4.5	0.511	0.7	1	0.725	4.5	0.3	0.5	0.6	0.8	2	5.3
6	Cole Aldrich	C	28	MIN	62	0	8.6	0.7	1.4	0.523	0	0	nan	0.7	1.4	0.523	0.2	0.4	0.682	2.5	0.4	0.4	0.4	0.3	1.4	1.7
7	LaMarcus Aldridge	PF	31	SAS	72	72	32.4	6.9	14.6	0.477	0.3	0.8	0.411	6.6	13.8	0.48	3.1	3.8	0.812	7.3	1.9	0.6	1.2	1.4	2.2	17.3
8	Lavoy Allen	PF	27	IND	61	5	14.3	1.3	2.8	0.458	0	0	0	1.3	2.7	0.461	0.4	0.5	0.697	3.6	0.9	0.3	0.4	0.5	1.3	2.9

If you don't know what the acronyms stand for, don't worry! We'll be working with the "PPG" column, which stands for "Points per game".

Assume all imports are correctly made.



3.1 Using the histogram above which analyzes points per game, answer the following questions:

a. Is it possible to find the percentage of players that scored between 12 and 15 points per game? Why or why not? What piece of information could help us answer this question?

b. Can we find the total number of players who averaged 20 or more points per game? What piece of information could help us answer this question?

Worksheet 3: Histograms, Functions

1 Histograms

Key Concepts

- Histograms are used to visualize the distribution of numerical data.
- We use bins to group numerical variables into intervals. They are inclusive of their lower bounds but exclusive of their upper bounds, which is often expressed as [lower, upper).
    - Bins need not always be the same size, so watch out for bins of unequal widths.
- The x-axis is in the units of the numerical variable that we are investigating.
- The y-axis, formally known as the density scale, measures the percent of data in the bin relative to the amount of space in the bin.
    - The reason we refer to it as density is that the y-axis can tell us how tightly clustered the data is in the bin.
- The area of a bin is equal to the percentage of data in the bin.
    - The larger the area of the bin, the more data lies in the bin!

Practice Problems

Throughout this first section, we'll be focusing on the following table: `nba.csv`. It describes the average statistics of NBA players for the 2016-2017 season. Ask your tutor if anything seems confusing.

The first few rows of the `nba` table look like this. There is one row for each player.

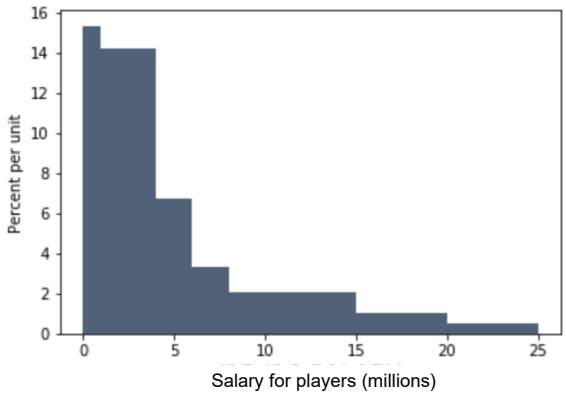
Rk	Player	Pos	Age	Tm	G	GS	MP	FG	FGA	FG%	3P	3PA	3P%	2P	2PA	2P%	FT	FTA	FT%	TRB	AST	STL	BLK	TOV	PF	PPG
1	Alex Abrines	SG	23	OKC	68	6	15.5	2	5	0.393	1.4	3.6	0.381	0.6	1.4	0.426	0.6	0.7	0.898	1.3	0.6	0.5	0.1	0.5	1.7	6
2	Quincy Acy	PF	26	TOT	38	1	14.7	1.8	4.5	0.412	1	2.4	0.411	0.9	2.1	0.413	1.2	1.6	0.75	3	0.5	0.4	0.4	0.6	1.8	5.8
3	Steven Adams	C	23	OKC	80	80	29.9	4.7	8.2	0.571	0	0	0	4.7	8.2	0.572	2	3.2	0.611	7.7	1.1	1.1	1	1.8	2.4	11.3
4	Arron Afflalo	SG	31	SAC	61	45	25.9	3	6.9	0.44	1	2.5	0.411	2	4.4	0.457	1.4	1.5	0.892	2	1.3	0.3	0.1	0.7	1.7	8.4
5	Alexis Ajinca	C	28	NOP	39	15	15	2.3	4.6	0.5	0	0.1	0	2.3	4.5	0.511	0.7	1	0.725	4.5	0.3	0.5	0.6	0.8	2	5.3
6	Cole Aldrich	C	28	MIN	62	0	8.6	0.7	1.4	0.523	0	0	nan	0.7	1.4	0.523	0.2	0.4	0.682	2.5	0.4	0.4	0.4	0.3	1.4	1.7
7	LaMarcus Aldridge	PF	31	SAS	72	72	32.4	6.9	14.6	0.477	0.3	0.8	0.411	6.6	13.8	0.48	3.1	3.8	0.812	7.3	1.9	0.6	1.2	1.4	2.2	17.3
8	Lavoy Allen	PF	27	IND	61	5	14.3	1.3	2.8	0.458	0	0	0	1.3	2.7	0.461	0.4	0.5	0.697	3.6	0.9	0.3	0.4	0.5	1.3	2.9

Assume all imports are correctly made.

1.1 NBA players must be at least 19 years old to play on a team. The oldest player that season was 40 years old. Create `age_bins` and assign it to an array of equally spaced bin values that describe the ages of NBA players with a bin width of 2.

1.2 Write code to create a histogram of the ages using the `age_bins` you just created.

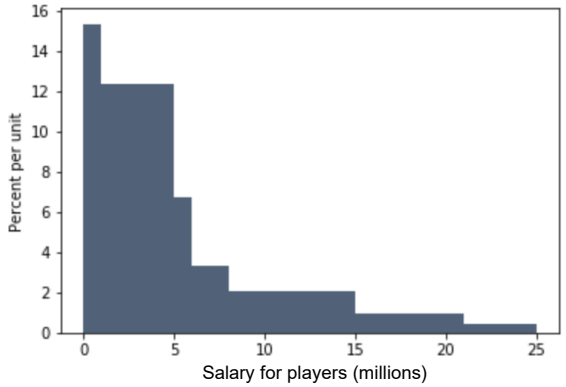
2.1 Let's now view the histogram below generated from the `nba_salaries.csv` table with the following code: `nba_salaries.hist(3,bins=make_array(0,1,4,6,8,15,20,25))`. Assume that all the players are represented in the histogram, and that the units for the salary data are in millions of dollars. Also note that this dataset contains 417 NBA Players. Answer the following questions with an arithmetic expression, or "Cannot answer".



a. What percentage of players in the dataset make between zero and one million dollars? What percentage of players make between one and four million dollars? Which bin has more players?

b. How many players make between 5 million and 6 million dollars?

2.2 Assume we have this second histogram generated using different bins: `nba_salaries.hist(3, bins=make_array(0,1,5,6,8,15,21,25))`

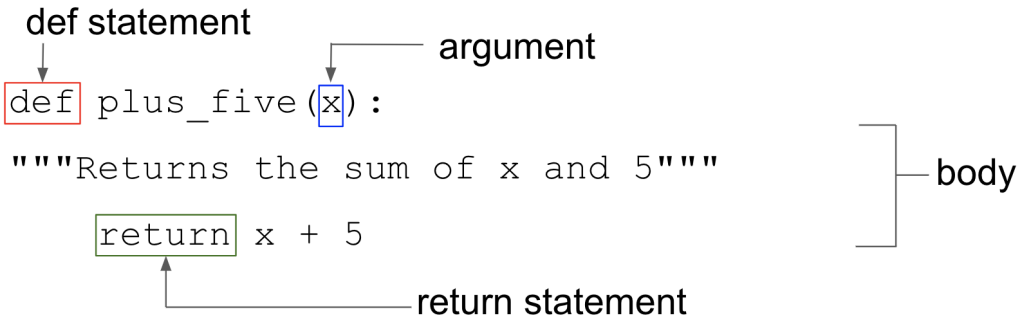


If you wrote "Cannot answer" for anything above, are you able to answer it now?

2 Functions

Key Concepts

Next, we will explore a tool that has been used many times already in this course: functions. We can define our own functions in order to give a name to a computational process that may be applied multiple times. The basic structure for a function is below:



Practice Problems

3.1 Define a function called `calculate_mean` that takes in an array of numbers and returns the average of the numbers in the array. Don't use the `np.mean` function!

```
def calculate_mean(array):
    sum_of_array = _____________________________________________________
    num_elements = _____________________________________________________
    return _____________________________________________________________
```

3.2 We have defined the function `calculate_statistics` below. Analyze the function and decipher what it does, then answer the questions below.

```
def calculate_statistics(array, multiplier):
   largest_num = max(array)                                    (1)
   smallest_num = min(array)                                   (2)
   array_average = calculate_mean(array)                       (3)
   stats_array = make_array(largest_num,                       (4)
                            smallest_num,
                                 array_average)
   final_array = stats_array*multiplier                        (5)
   return final_array                                          (6)
```

Suppose you execute the line of code below in a blank cell. Answer the questions below.

```
statistics = calculate_statistics(make_array(5, 10, 15, 20), 2)
```

a. After this function is called, what does `largest_num` get assigned to?

b. What does `array_average` get assigned to?

c. What does `stats_array` get assigned to?

d. What does `final_array` get assigned to?

e. What does the function return? What is its type? (i.e. `int`, `string`)

f. After the line of code is executed, what would happen if we tried to display the value of `largest_num`?

g. Finally, if we ran `calculate_mean(statistics)` after `statistics` is assigned, what would we get back as our output?

Worksheet 4: Conditionals, Iteration

1 Conditional Statements and Iteration

Key Concepts

Conditional Statements

We can use conditional statements to write code and create functions that perform different operations based on certain conditions. As a reference, here is how conditional statements work:

```
if x > 10:
   Do something
elif x > 5:
   Do something
else:
   Do something
```

Iteration

For loops in Python can potentially allow us to do two different operations. First, they allow us to iterate through arrays, manipulating each element as we wish. Alternatively, we can use for loops to repeat lines of code many times. Examples of how for loops can be used are below.

```
for item in some_array:
   print(item)
```

or

```
for i in np.arange(1000):
   print("Hello")
```

Practice Problems

Question 1. Examine the function, then answer the questions below. It has been written with a purposely vague name and arguments!

```
def mystery_function(x):
   if (x > 0):
         return "Positive"
   elif (x < 0):
         return "Negative"
   else:
         return "Neither"
```

1.1 What would `mystery_function(10)` return?

1.2 What does `mystery_function(-1)` return?

1.3 What does `mystery_function(0)` return?

Question 2. The for loop statement below stores the length of each name in `names` in a new array called `lengths`.

```
lengths = make_array()
names = make_array('Bob', 'Sarah', 'Michael', 'Sam')

for name in names:
   lengths = np.append(lengths, len(name))
```

2.1 For each iteration below, fill in the value of `name` as well as what `lengths` looks like.

Iteration 1: `name = _____ , lengths = _____`
Iteration 2: `name = _____ , lengths = _____`
Iteration 3: `name = _____ , lengths = _____`
Iteration 4: `name = _____ , lengths = _____`

2.2 Now, let's say that instead of storing lengths, we want to store the name as long as the length of the name is greater than 4. Fill in the following for loop statement such that `longer` contains these names.

```
longer = make_array()

for name in _____________________________:
   if ____________________________:
         longer = ____________________________________
```

2.3 What names would `longer` contain after the for loop executes?

2.4 Finally, look at this last for loop below. What values does `i` take on throughout? How is `i` used as compared to the way `name` is used in the previous for loops?

```
counter = 0
for i in np.arange(1000):
   counter = counter + 1
```

Question 3. Suppose you have an array called `salaries`, containing the salary information of 5 individuals. You would like to determine what percentage of the total salaries each individual's salary comprises. You want to output an array, `proportion` where the ith element of proportion corresponds to what percentage of the total salary salary.item(i) is.

For example, if `salaries` was equal to an array `[1,2,3,1,3]`, then `proportion.item(0)` would be 0.1.

3.1 Your friend writes some code, but it doesn't work! Find the error that your friend made. What would the code output if executed as is? How would you fix it?

```
salaries = make_array(25, 50, 100, 25, 100)
total = sum(salaries)

for salary in salaries:
   proportion = make_array()
   percentage = salary/total
   proportion = np.append(proportion, percentage)
```

3.2 You fix the error described above, but in doing so, break something else. Again, find the error in the code below. What would the code output if executed as is? How would you fix it?

```
salaries = make_array(25, 50, 100, 25, 100)
total = sum(salaries)
proportion = make_array()

for salary in salaries:
   percentage = salary/total
   np.append(proportion, percentage)
```

Worksheet 5: Sampling and Hypothesis Testing

1 Sampling

Key Concepts

Population vs. Sample

In data science, we often want to be able to make a general statement about a **population** of individuals. Unfortunately, resource constraints generally prevent scientists from having access to data about entire populations of individuals. For that reason, we examine parts of the population called **samples**. Our goal is to infer some characteristics of the population, called **population parameters** from the study of our sample. In many cases, we are interested in estimating these parameters using **sample statistics**, or quantities that we measure from a sample of the population.

For example, you may be interested in knowing the percentage of all eligible voters who are registered to vote for the upcoming election. Since asking everyone in the U.S. if they have registered to vote is clearly infeasible, we will have to take a sample.

Sometimes, we will want to sample from a pre-existing table. To do so, we can use the following table method:

```
tbl.sample(sample_size)
```

In other cases, we may have an array we need to sample from. In this case, we can use the following function:

```
np.random.choice(array, sample_size)
```

Practice Problems

1.1 Let's use the example of rolling a fair die. Remember: rolling a die is always sampling "with replacement".

a) What is the probability that you will roll a 5? Is this an empirical or a theoretical probability? Is there a relationship between the two?

b) Complete the function `roll_die`, which takes in no arguments and uses `dice` table to the right to roll a dice a single time and returns the value randomly picked.

Side
1
2
3
4
5
6

the
that is

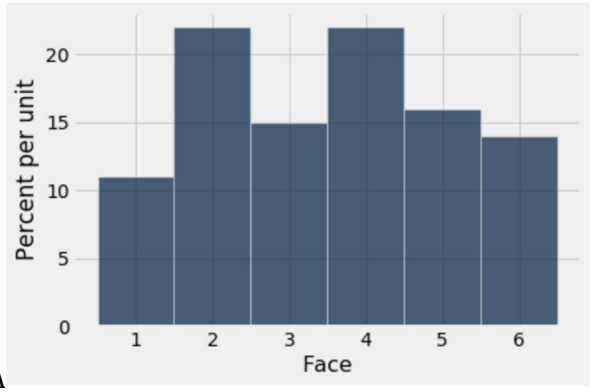
```
def roll_die():
```

c) Simulate rolling a die 10 times and store the results in an array called `simulated_rolls`.

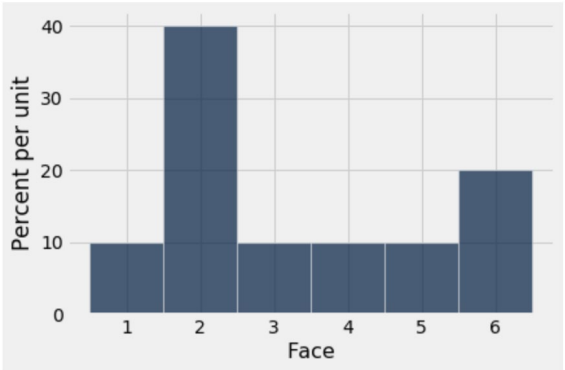
```
simulated_rolls = make_array()

for i in _____________________:
      face = ____________________________________
      simulated_rolls = ____________________________________
```

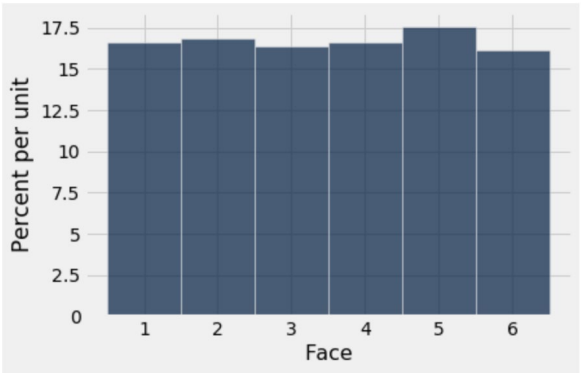
d) We've generated histograms of dice roll results for samples of size 10, 100, 1000, and 10,000 below. Which histograms correspond to which sample sizes, and why?

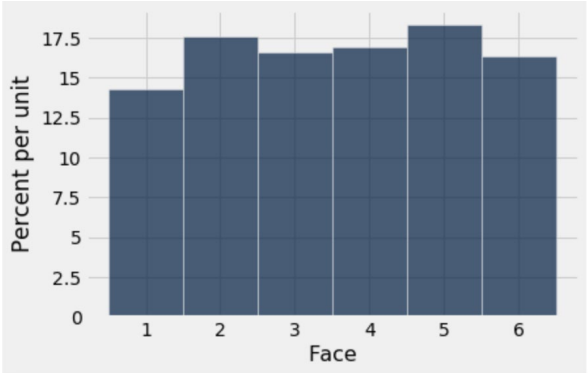
A



B



C 

D 

2 Hypothesis Testing

Key Concepts

Suppose you flip a (presumably) fair coin 20 times, and see that the coin comes up heads 18 out of the 20 flips of the coin. This seems strange to you, as you previously believed that the coin is fair. A natural question to ask would be - was the 18 heads in 20 flips due to random chance? Or was it due to something other than random chance?

Hypothesis testing uses the power of computation to allow us to answer the question of "Was this strange event due to random chance?" in a scientific and consistent manner.

Practice Problems

2.1 Suppose you are flipping thumbtacks, and thumbtacks always either land pointing up or pointing down. You flip a thumbtack 60 times, and observe the thumbtack land pointing down 45 times. Your friend tells you that a thumbtack lands down with a 2/3 chance, and lands up with a 1/3 chance.

a. Does the thumbtack that you are flipping seem consistent with your friend's model?

b. Complete the function `flip_thumbtack`, which takes in no arguments and randomly flips a thumbtack 60 times. The thumbtack lands down with probability 2/3 and lands pointing up with probability 1/3. The function returns the number of pointing down results out of the 60 tosses.

```
def flip_thumbtack():
    probabilities = _____________________________________________
    proportions = sample_proportions(_____________________________)
```

```
proportion_down = _____________________________________________
return _____________________________________________________
```

2.2 Suppose you want to leave your breakfast choices up to chance! You have a cabinet of 4 different cereal brands: Cheerios, Lucky Charms, Fruit Loops, and Cocoa Puffs. Suppose you randomly pick 4 cereal boxes *with replacement*.

a. What is the probability that you pick four unique brands of cereal?

b. What is the probability that you don't pick Cheerios?

2.3 In the Netherlands, all men take a military preinduction exam at age 18. The exam includes an intelligence test known as "Raven's progressive matrices" and includes questions about demographic variables like family size. A study was done in 1968, relating the test scores of 18-year-old men to the number of their brothers and sisters. The records of all exams taken in 1968 were used.[1]

a) What is the population of the study? What is the sample used in the study?

b) Is there a need to apply inference techniques to predict the mean score, max score, etc.? Why or why not?

[1] Taken from Statistics Fourth Edition by Friedman, Pisani and Purves

Worksheet 6: Midterm Review

1. Table Practice

The `nba` table contains data from the 2017-2018 season for every active player. Each row represents statistics over the whole season. In this table, percentages are expressed as decimals.

Player	Age	Team	Games	Minutes	FG	FGA	FG%	3P	Rebounds	Assists	Steals	Blocks	Turnovers	Points
Alex Abrines	24	OKC	75	1134	115	291	0.395	84	114	28	38	8	25	353
Quincy Acy	27	BRK	70	1359	130	365	0.356	102	256	57	33	29	60	411
Steven Adams	24	OKC	76	2487	448	712	0.629	0	685	88	92	78	128	1056
Bam Adebayo	20	MIA	69	1368	174	340	0.512	0	381	101	32	41	66	477
Arron Afflalo	32	ORL	53	682	65	162	0.401	27	66	30	4	9	21	179
Cole Aldrich	29	MIN	21	49	5	15	0.333	0	15	3	2	1	1	12
LaMarcus Aldridge	32	SAS	75	2509	687	1347	0.51	27	635	152	43	90	111	1735
Jarrett Allen	19	BRK	72	1441	234	397	0.589	5	388	49	28	88	82	587
Kadeem Allen	25	BOS	18	107	6	22	0.273	0	11	12	3	2	9	19
Tony Allen	36	NOP	22	273	44	91	0.484	4	47	9	11	3	19	103

a) eFG% is an advanced statistic commonly used over FG% (field goal percentage). Calculate eFG% using the formula (FG + 0.5*3P)/FGA. Make sure to use `FG` and not `FG%`.
Once it's been calculated, append the values as the column `eFG%` to this table.

```
numerator = nba._____(_____) + _____*nba._____(_____)
denominator = _____._____(_____)
nba_efg = nba._____(_____, numerator / denominator)
```

b) Find the team with the lowest average eFG% (return the name only)

```
by_team = nba_efg._____(_____, _____)
by_team.sort(_____)._____(_____)._____(_____)
```

c) What proportion of points scored were by players who had an eFG% above 60%? The variable `answer` should be your final proportion.

```
more_than_sixty = _____(nba_efg._____(eFG%,
                  _____._____(_____)).column(_____))
total = _____(nba_efg._____("Points"))
answer = more_than_sixty / total
```

2. Coding Practice

a) Suppose a broken candy machine dispenses sweet candy 99% of the time and sour candy otherwise. What is the chance you find at least 1 sour candy in 50 candies dispensed randomly from the machine? Let `candies` be an array with the first element containing the probability of picking a sweet candy and

the second element being the probability of picking sour candy. Use a simulation to estimate the probability of finding at least 1 sour candy in 50 candies dispensed.

```
candies = make_array(0.99, 0.01)
sour = ___________
for i in ___________________(5000):
chosen_candies_prop = sample_proportions(_____________________________)
sour_prop = ____________________________________________________________
if _____________________________:
      sour = sour + 1
chance_of_at_least_one = ___________/5000
```

b) What is the exact probability of finding at least 1 sour candy in 50 candies dispensed from the broken candy machine?

Hint: think about the complement rule!

3. Defining Functions

Suppose you have a table `dinners`, which contains a row for every dinner eaten at a given restaurant for a week. `dinners` has a column "Day" of strings corresponding to which day of the week the dinner was eaten and a column "Subtotal" of integers containing costs without tips.

a) Define a function called `compute_tip` that given a total bill as an integer, returns a 20% tip calculated from the bill.

```
def compute_tip(bill):
   return ____________________________________________
```

b) Add a new column to the `dinners` table called "Tip" that is the 20% tip for each bill in the "Subtotal" column and name the resulting table `dinners_with_tip`.

```
dinners_with_tip = dinners._____________________("Tip",
             dinners._________________(___________________, _____________))
```

c) Write code to calculate the day of the week with the lowest average cost (not including tip).

```
by_day = dinners._____(_____, _____)

by_day.sort(_____)._____(_____).item(0)
```

4. Probability

Your good friend Gary is playing with a fair 6-sided die.
a) Gary rolls the die 10 times and rolls a 6 every time. What is the probability of that event occurring?

b) Then, Gary rolls the die twice. What is the probability he rolls a 1 on the first roll and a 2 on the second roll?

c) Suppose Gary rolls the die 5 times - what is the probability he rolls at least one 3?

d) If you are considering whether the die is fair, do the results of the throws represent numerical or categorical outcomes?

5. Hypothesis Testing

Gary rolls the die another 10 times and rolls 7 sixes, 2 fours, and 1 two. I suspect Gary is using an unfair die and I want to do a hypothesis test to check this.

a) Specify a null and alternative hypothesis.

b) What test statistic would you choose to compare the null distribution above with what you simulate repeatedly to test whether the die is fair? Explain.

c) Calculate the obs_test_stat.

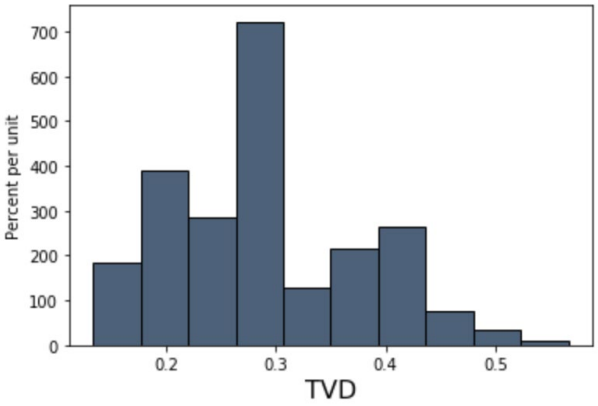
```
fair_die = make_array(1/6, 1/6, 1/6, 1/6, 1/6, 1/6)
gary_die = ___________(___, ___, ___, ___, ___, ___)

obs_test_stat = 0.5 * _____(_____(____________________ - ________________))
```

d) Fill in the code below to simulate the distribution of faces if we roll a fair die 10 times and calculate one test statistic.

```
def calculate_test_stat():
    fair_die = make_array(1/6, 1/6, 1/6, 1/6, 1/6, 1/6)

    simulated_dist = ____________________(________,_________________________)

    test_stat = 0.5 * _____(_____(____________________ - ____________________))

    return test_stat
```

e) Fill in the code below to simulate 10000 test statistics and generate the following histogram.

```
test_stats = _______________

for i in _________________:

    one_test_stat = ____________________

    test_stats = ___________(_____________, _____________)

Table().with_column(_____, _____).hist(_____)
```



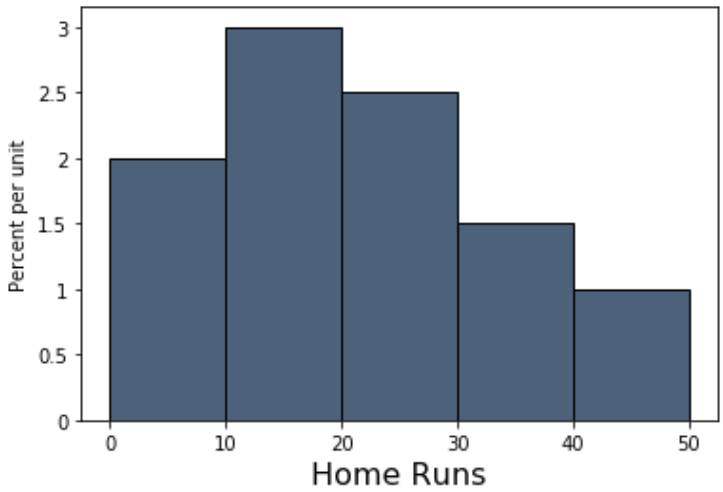
f) Fill in the code below to calculate the p-value.

```
p_value = _________(____________________)/____(_________)
```

g) If `p_value` turns out to be 0.004 and we use a p-value cutoff of 0.05, what conclusion do we draw from this test?

6. Histograms

The following is a histogram of the number of home runs hit by MLB players in the 2019 season.



Answer the following questions using the histogram above. If it is not possible to compute the answer, write "Not Possible" and explain why you cannot calculate the answer.

a) Find the percent of players who hit between 10 and 20 (not inclusive) home runs in 2019.

b) Find the number of players who hit more than 30 home runs in 2019.

c) What percent of players hit at least 20 home runs?

d) How many players hit between 25 and 30 home runs?

e) We decide that we want more information about the players that hit between 10 and 20 home runs, so we split the [10,20) bin into two bins: a [10, 15) bin and a [15, 20) bin. We find that 20% of players hit between 10 and 15 home runs. What is the height of this new [15, 20) bin?

Worksheet 7: Sample Means, Center/Spread, Normal Distribution

1 Mean and Median

Key Concepts

Mean: Definition

The average, or mean, of a collection of numbers is the sum of all the elements divided by the total number of elements in the collection.

Median: Definition

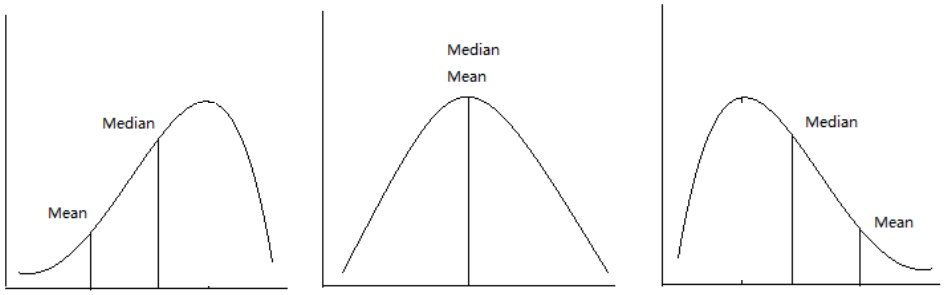
The median is the 50th percentile of a collection of numbers. It is the "middle" element.

Properties of the Mean and Median

- They mean and median aren't necessary elements of the set of numbers.
- They might not be an integer even if all the elements of the collection are integers.
- If the collection consists of values measured in specified units, then it has the same units too.

Mean vs. Median

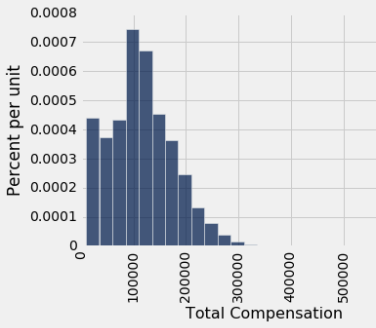
The median is always the midpoint of the data, while the mean is affected by the magnitude of the data points. For example, if the data is concentrated to the right with fewer values on the left, the mean is dragged to the left by those tail values.



Practice Problems

1.1 Suppose a set of numbers has mean value 15 and median value 20. Is the distribution of the values in the data skewed *left* or skewed *right*?

1.2 In the graph to the right, is the mean or the median larger?



1.3 Suppose you have an array containing three 18s, seven 11s, and a 74.

a. Write an arithmetic expression to calculate the mean of the array. How does the 74 affect the histogram?

b. Now suppose we replace the 74 with 350. How does this affect the mean? How about the median?

2 Variability

Key Concepts

Calculating Variance and SD

SD: "Root mean squared deviation from average"

5 4 3 2 1

Assume `dist = [2, 4, 6, 8, 10]`

- First, find the average of the distribution.
  - `average = 6`
- Next, find the difference between each number in the distribution and the average.
  - `differences = [-4, -2, 0, 2, 4]`
- Square each difference (so there are no negatives).
  - `squared_differences = [16, 4, 0, 4, 16]`
- Now take the mean of all the squared differences. (Variance)
  - `mean_squared_differences = 8`
- Take the square root of that mean. (Standard Deviation)
  - `root_mean_squared_differences =` $\sqrt{8}$

Alternatively, you can also use `np.std(array)` to calculate the standard deviation!

Standard Units

To convert a value to standard units (a unitless measure), first find how far it is from the average of the distribution, and then compare that deviation with the standard deviation of the distribution.

$$z = \frac{\text{value} - \text{average}}{\text{SD}}$$

Practice Problems

2.1 Write code to convert the delay times in column "Delay" from the `united` table at right to standard units. Name the array of converted times `delay_standard`.

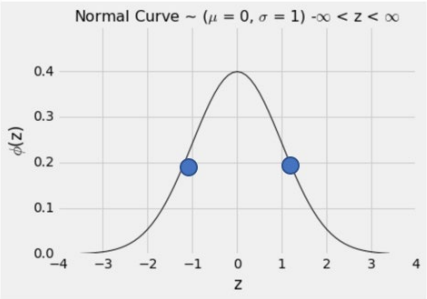
Date	Flight Number	Destination	Delay
6/21/15	1964	SEA	580
6/22/15	300	HNL	537
6/21/15	1149	IAD	508
6/20/15	353	ORD	505
8/23/15	1589	ORD	458

3 SD and Normal Curve

Key Concepts

Overview

Here is the standard normal curve (mean = 0, SD = 1) and some of its properties:



- The total area under the curve is 100%.
- The curve is symmetric about 0, with its mean and median both equal to 0.
- If a variable has this distribution, its SD is 1. The standard normal curve is one of the few distributions that has a SD so clearly identifiable on the histogram.

Chebyshev's Bounds: The table on the right uses Chebyshev's inequality to calculate the following proportion of values that fall within k SDs of the mean. Remember that Chebyshev's bound works for **ALL** distributions, which is why it is a weaker bound.

Range	Proportion
average ± 2 SDs	at least 1 - 1/4 (75%)
average ± 3 SDs	at least 1 - 1/9 (88.888…%)
average ± 4 SDs	at least 1 - 1/16 (93.75%)
average ± 5 SDs	at least 1 - 1/25 (96%)

The table below shows the Chebyshev bounds for the normal distribution.

Percent in Range	Normal Distribution: Approximation
average ± 1 SD	about 68%
average ± 2 SDs	about 95%
average ± 3 SDs	about 99.73%

Practice Problem

3.1 Vehicle speeds on a highway are normally distributed with mean 90 mph and SD 10 mph. Using the table above, what is the approximate probability that a randomly chosen car is going more than 100 mph?

Hint: Remember that the total area under the normal curve is 1, and that the area under a region of the curve represents the proportion of total data that falls in that region.

4 Central Limit Theorem

Key Concepts

Overview

The Central Limit Theorem says that the probability distribution of the **sum or average of a large random sample drawn with replacement will be roughly normal**, regardless of the distribution of the population from which the sample is drawn.

Practice Problems

4.1 Suppose you simulate the proportion of purple-flowered plants in a sample of 200 plants (from Mendel's 75% purple- and 25% white-flower plant population) using `sample_proportions` 1000 times. Then, you plotted distribution of the proportion of purple-flowered plants from each of the 1000 trials. What would this distribution look like? Where would the distribution be centered?

4.2 What would it look like if we used a sample size of 800 instead?

5 Variability of the Sample Mean

Key Concepts

The SD of the Sample Mean

$$\text{SD of all possible sample means} = \frac{\text{Population SD}}{\sqrt{\text{sample size}}}$$

This is the standard deviation of the averages of all the possible samples that could be drawn. **It measures roughly how far off the sample means are from the population mean.** The smaller the SD, the more accurate the estimate.

Practice Problems

5.1 As sample size increases, what happens to the distribution of the sample mean? Does it become narrower or wider? Where is it centered?

5.2 Does population size affect the variability of the sample mean?

5.3 If you had a sample size of 100, but wanted to increase accuracy by a factor of 4, what should the new sample size be?

Worksheet 8: Designing Experiments, More CLT

1 The Variability of the Sample Mean

Key Concepts

The SD of the Sample Mean

$$\text{SD of all possible sample means} = \frac{\text{Population SD}}{\sqrt{\text{sample size}}}$$

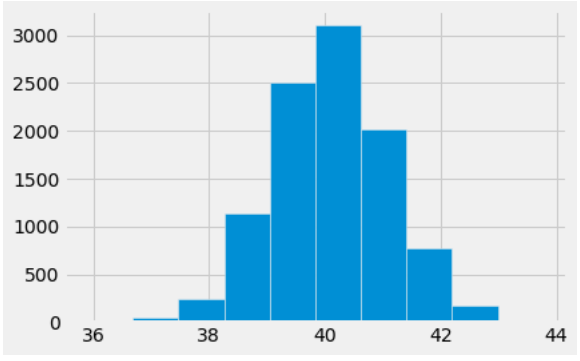
This is the standard deviation of the averages of all the possible samples that could be drawn. **It measures roughly how far off the sample means are from the population mean.** The smaller the SD, the more accurate the estimate.

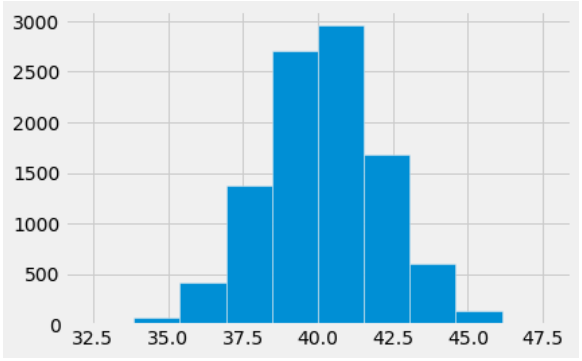
Proportions are Means

- Proportions are the means of a list of 1s and 0s (they tell you the fraction of 1s in the list)--this means that the CLT applies to them!
- Oftentimes, you'll work with data where the population is binary. For example, you might have polling data where the answer to a question is "Yes" or "No". You might want to estimate the sample proportion of "Yes" votes.
- The SD of all possible sample proportions, calculated using the formula above, is **at most 0.5 regardless of the proportion of 1s in the population**.

Practice Problems

1.1 Suppose we have a population with a mean of 40 and an SD of 10. One of the histograms below is an empirical distribution of the means of 10000 bootstrap resamples each of size 100 from the population. Which histogram?





1.2 Fill in the blank: Based on the population from the previous question, there is a _____% chance that a random resample has a mean that lies within the range [38, 42].

1.3 Suppose a redwood forest has trees whose average height are 200 feet with an SD of 30 feet. A random sample of 400 trees is taken. Fill in the blank: There is a 68% chance that the average height of the sample lies within the range 200 plus or minus _____.

2 Designing Experiments and Choosing Sample Size

Key Concepts

Choosing the Sample Size of an Experiment

- Sometimes, you'll need to conduct an experiment and estimate a population parameter (like a mean) up to a certain accuracy--this accuracy is usually measured by the width of the confidence interval.
- Ultimately this means that we want to limit the variability of our estimate. We know from the CLT that the variability of the sample mean is affected by the sample size!
- For a normal distribution, the "middle 95%" is within 2 SDs of the mean. We can use this information to create a confidence interval: the center ± 2 SDs.
- Then, the width of this confidence interval is:

$$\text{width of 95\% CI} = 4 \times \frac{\text{SD of the population}}{\sqrt{\text{sample size}}}$$

- The term being multiplied is the SD of all the sample means--if the bounds of our 95% confidence interval are 2 SDs to the left of the mean and 2 SDs to the right, then the width is 4 SDs total!
- We can use this equation above to calculate the sample size we need for an experiment.

Practice Problems

Let's say you want to poll the population of UC Berkeley students to ask whether they like vanilla ice cream or chocolate ice cream. You can only take a sample, but you want to estimate the population proportion of students who like vanilla ice cream. Let's say you need your estimate to have a confidence interval width of at most 0.05.

2.1 Suppose the population SD of the proportion of students who like vanilla ice cream is 0.1. What sample size do you need to achieve a 95% confidence interval width of **at most** 0.05?

2.2 Is it possible to calculate what sample size you need if you don't know the population SD? If not, can we bound what the population SD could be?

2.3 Suppose you **do not** know the population SD of students who like vanilla ice cream. What sample size do you need to achieve a 95% confidence interval of width **at most** 0.05?

3 (Optional) Confidence Interval Review

Practice Problems

Tonight is the Monster Mash. We're trying to determine the median scariness level of ghosts. We are given a sample of ghosts in the form of a one column table, `spooky_sample`, that contains 200 numbers each of which describe how scary a ghost is on a scale from 0 to 10. You can assume that the sample is a simple random sample from the population of all ghosts.

3.1 Fill in the code below to create a function that computes a 95% confidence interval for the median scariness level in the population of ghosts. Assume `spooky_sample` is a one column table with scariness levels of the ghosts in our sample.

```
def candy_cornfidence_interval(spooky_sample, replications):
   result_medians = ____________________
      for i in ______________________________:
         resample_median = ___________________________________
         median = ____________________________________________
         result_medians = ____________________________________
   left_end = ____________________________
   right_end = ____________________________
   return ___________________________
```

3.2 If we run the function you wrote above multiple times, will it always return the same interval? Why or why not?

3.3 If we consider the population of all ghosts, there exists a median scariness level. We call this the "true median scariness level" of the population. Recall that since we don't have access to the population, we don't have access to the true median scariness level either.
If we were to compute 100 confidence intervals with the function from 3.1, how many of those confidence intervals would we expect to capture the true median scariness level?

3.4 If we picked out one of the 100 confidence intervals from the previous question and found that it was [5.6, 6.8], what is the probability that this interval contains the true median scariness level?

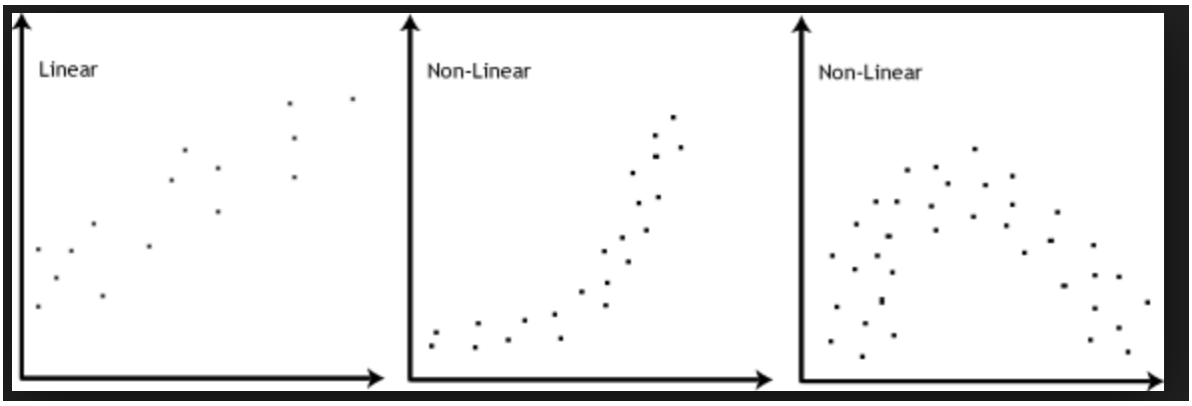
Worksheet 9: Correlation, Regression

1 Correlation

Key Concepts

Association

- Refers to any relationship between two variables. It does not have to be linear. For instance, in the plots below, only the first graph demonstrates a linear association.



Correlation

- Denotes a linear association between two variables.
- The *correlation coefficient* quantitatively measures the strength as well as the direction of the linear relationship between two variables.
- The correlation coefficient is denoted as *r*, a number between -1 and 1
    - Strength: how clustered the scatter plot is around a straight line. If the plot is highly clustered, the absolute value of *r* is closer to 1
    - Direction: if y increases as x increases, *r* is positive. If y decreases as x increases, *r* is negative.

Standard Units

- Allows us to quantify the relationship between two variables on different scales
- Converting to Standard Units
    - variable_su = (variable – mean) / SD

A Formula for r

- The average of the product of *x* and *y*, when both variables are measured in standard units.
- r = 1 if the scatter diagram is a perfect straight line sloping upwards, and r = −1 if the scatter diagram is a perfect straight line sloping downwards.
- r is *a number without units*. This is because it is computed with standard units, which have no units.
- r is *unaffected by changing the units* on either axis. This too is because r is based on standard units.

- r is *unaffected by switching the axes*. This is because it is the sum of products of standard units; xy = yx. More intuitively, since correlation is a measure of spread around a line, switching the axes won't change the spread around the line.

Practice Problems

1.1 The following table, taters, depicts the number of tater tots a person has ate, along with a number that quantifies their satisfaction, which is a number that goes from 0 to 10.

Tater Tots Consumed	Satisfaction
1	8
10	3
4	7
3	10
7	6
3	8

a) Complete the function `standard_units` which takes in an array `num_array` and returns the same array in standard units.

```
def standard_units(num_array):
        arr_mean = ____________________________________
        arr_sd = ____________________________________
        return ____________________________________
```

b)Fill in the blanks to define a function correlation that finds the correlation from a table. It takes in three arguments: a table, `tbl`, and two column indices, `x` and `y`.

Hint: Use the `standard_units` function defined above!

```
def correlation(tbl, x, y):
        su_x = _____________________(_____________________________)
        su_y = _____________________(_____________________________)
        return _________________(_____________________________________)
```

c) Calculate r by using the correlation function.

```
correlation(_________________, _________________, _________________)
```

d) Suppose that we calculated a value of *r* to be equal to -0.879. What can you conclude about the association between the number of tater tots consumed and a person's satisfaction?

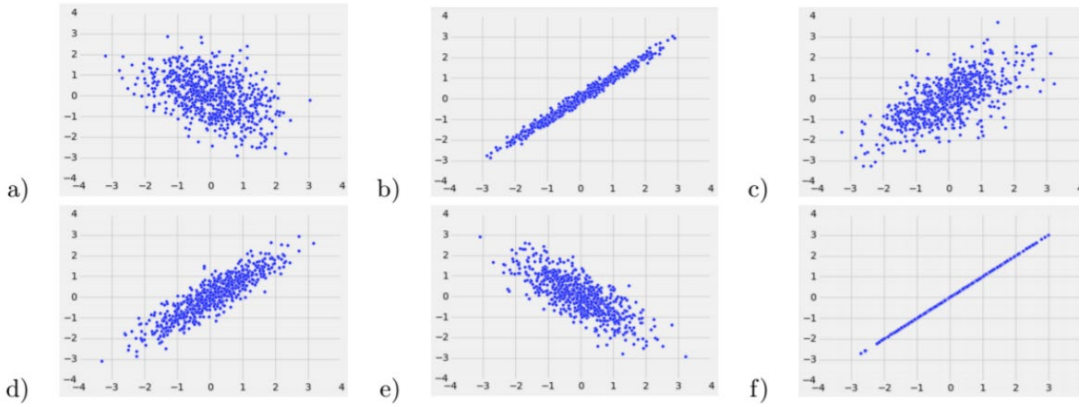
1.2 True or False?

a. A high value of *r* shows that a change in *x* causes a change in *y*.

b. If we switch the axes of a plot, the correlation coefficient will not change.

c. Suppose that we calculated a value of *r* to be equal to .83. We should conclude that eating taters is indeed correlated with satisfaction.

1.3 Answer the following questions about the plots below.



a. Order the scatter plots above in from least correlated to most correlated.

b. Which plots have a positive correlation coefficient? Negative correlation coefficient?

2 Regression

Key Concepts

Correlation Coefficient

- The equation for the data's regression line can be calculated using *r*. Recall that the equation of a line is y = slope · x + intercept.

Standard Units

- Graphically, the scatterplot and regression line look the same whether x and y are in standard units or their original units.
- Calculating the regression line when *x* and *y* are in standard units:
  - estimate of *y* = *r* · the given *x*, where *x* and *y* are in standard units

Calculating the regression line when *x* and *y* are in original units:

1) Calculate the correlation coefficient, *r*.
2) Calculate the slope.

slope of the regression line = $r * \frac{\text{standard deviation of } y}{\text{standard deviation of } x}$

3) Calculate the intercept.

intercept of the regression line = mean of y − slope * mean of x

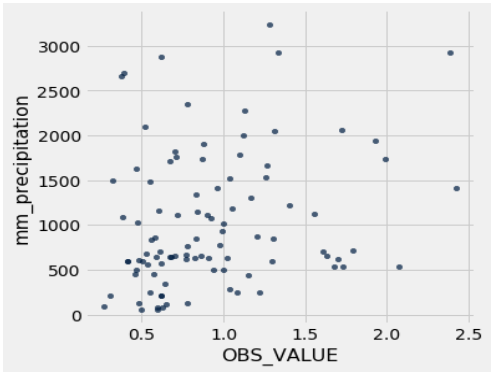
Practice Problems

The `water` table contains one row per country with data from 2014. The `OBS_VALUE` column represents the approximate price ranking of a 1.5 liter bottle of mineral water in that country, and the `mm_precipitation` column represents the average precipitation in that country (in millimeters).

COUNTRY	OBS_VALUE	mm_precipitation
Albania	0.55	1485
Algeria	0.27	89
Angola	1	1010
Argentina	1.29667	591
Armenia	0.5325	562
Australia	2.07302	534
Austria	0.72	1110
Azerbaijan	0.576	447
Bangladesh	0.374	2666
Belarus	0.7675	618

... (89 rows omitted)

Expression	Values
np.average(water.column('OBS_VALUE'))	0.919016
np.std(water.column('OBS_VALUE'))	0.464763
np.average(water.column('mm_precipitation'))	1010.4
np.std(water.column('mm_precipitation'))	752.475
correlation(water, 'OBS_VALUE', 'mm_precipitation')	0.262079



2.1 What is the value of `correlation(water, 'mm_precipitation', 'OBS_VALUE')`?

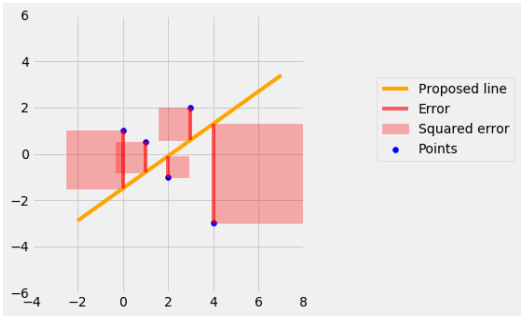
2.2 Write an equation for the regression line of the data in the water table, using `OBS_VALUE` as *y* using the `mm_precipitation` as *x*.

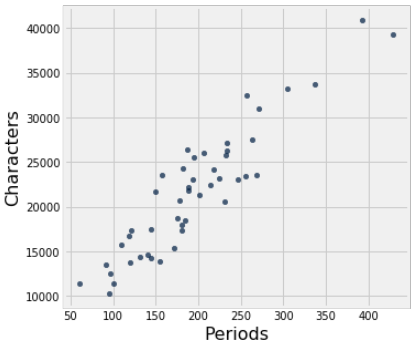
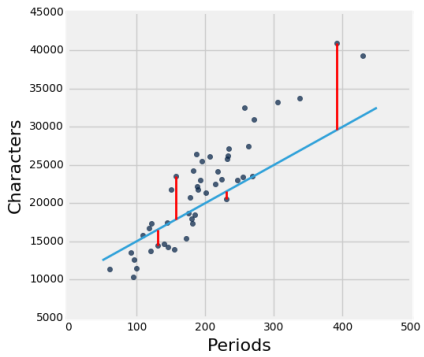
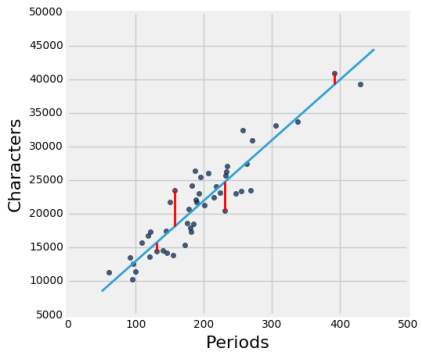
2.3 Using the regression line equation above, what would we expect the `OBS_VALUE` to be in 2014 for a country that had an average of 700 mm of precipitation?

3 Root Mean Squared Errors (RMSE)

Key Concepts

- Root Mean Squared Error is the square root of the average of the squared errors
- RMSE = $\sqrt{(\text{error}_0^2 + \text{error}_1^2 + \ldots + \text{error}_n^2) / n}$, where n is the number of points in our dataset and each error = observed value − predicted value



Original Plot	Poor Fit (High RMSE)	Best Fit (Lowest RMSE)
		

Practice Problems

3.1 Write a function that returns the RMSE of an array of observed values if the predicted values are given by an array. The two arrays have the same length.

```
def RMSE(observed, predicted):
    residual = _____________________________________
    squared_residuals = ____________________________________________
    squared_resid_avg = ____________________________________________
    return _________________________________________
```

3.2 In the calculation of root mean squared error, why is it important for us to square the residual before taking the sum?

Worksheet 10: Residuals and Regression Inference

1 Residuals

Key Concepts

Definition/Properties

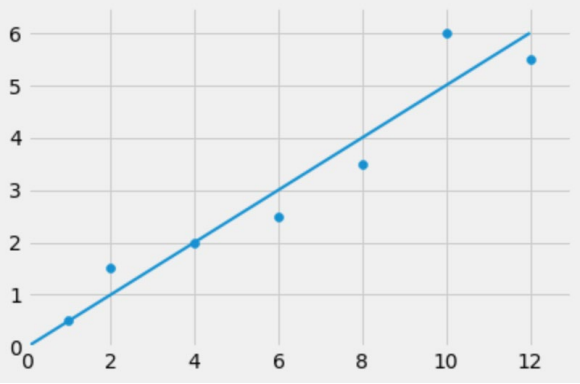
- A residual is the difference between an observed value and its corresponding regression estimate.
- Visually, residuals are also the vertical difference between each observed point and its corresponding estimated point.
- The larger the absolute value of the residual, the further away our estimate is from our actual data point. If the estimates are equal to our data points, then all of our residuals will be equal to 0.
- residual = y − estimated value of y = y − height of regression line at x
- The sum of all residuals is 0.
- SD of residuals = $\sqrt{1 - \square^2}$ ∗SD of y

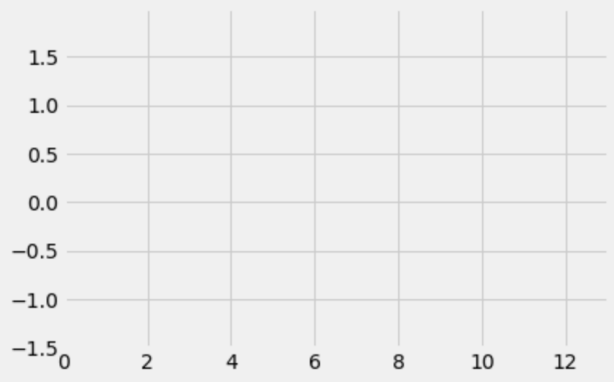
The relationship between Residual Plots and Regression

- Residual plots are used to visually diagnose how well our regression line fits the data.
- First of all, the goal of least squares is to choose a line of best fit that minimizes error. We can use least squares to help us calculate the best slope and intercept to fit our data.
- If linear regression is a good method to use, then there will be *no pattern* in the our plot of residuals.

Practice Problems

1.1 Given the following regression line, draw an approximate residual plot.





1.2 Answer whether the following questions are True or False.

a) If we perform linear regression on two variables, the residual plot never has a trend.

b) No matter what the shape of the scatter diagram, the average of the residuals is 0.

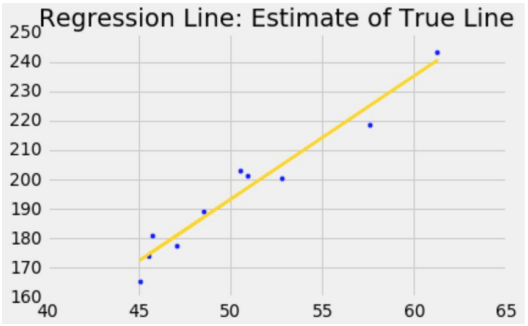
c) No matter what the shape of the scatter plot, the SD of the residuals is less than or equal to the SD of the true y values.

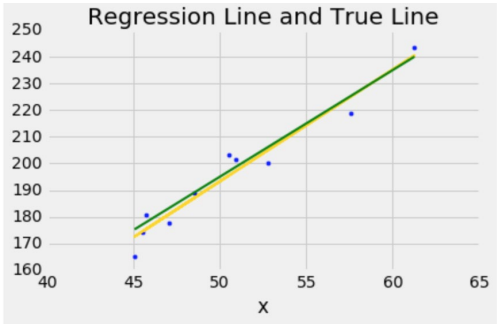
2 Regression Inference

Key Concepts

Inference for the True Slope

- Thinking about the bigger picture, let's assume that we have a dataset that is completely linear and begin pushing the points away from the line at random, symmetrically on both sides. You end up with a data set that is clustered around the "true line."





Key Mathematical Definitions

- correlation coefficient $r = average(x_{standard\ units} * y_{standard\ units})$
- *slope* = $r * \frac{SD\ of\ Y}{SD\ of\ X}$
- *intercept = average of Y – slope * average of X*

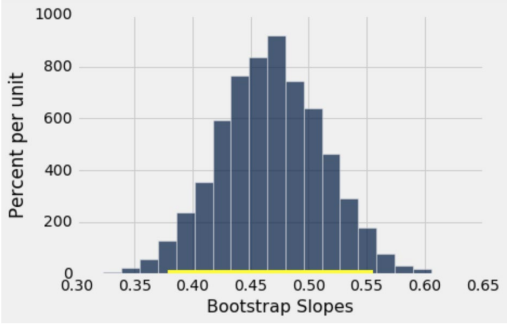
Practice Problems

2.1 True or False:

a) The regression line's *x* and *y* values are always measured in standard units.

Previously in this class, you all constructed confidence intervals using the bootstrap to predict population parameters, such as the population mean. When investigating the correlation between two variables, you can actually construct a confidence interval for the slope of the best fit line using the same method!

Performing this calculation can allow you to determine if you believe there actually exists a correlation between two variables, or if the relationship you observe between the variables is due to random variation in the sample.



2.2 Why do we need to bootstrap slopes to test if a true slope is 0 or not? Why can't we just tell from the sample slope we get?

Let's try to estimate a real slope using this bootstrap technique. We'll be pulling an example from the textbook so you can look back at it for future reference. We have a dataset on babies' birth weights and corresponding gestational days (days in the womb before birth).
One might guess that the longer a baby is in the womb, the heavier it is when it is born - since it has more time to grow.

The table `baby` is as follows:

Gestational Days	Birth Weight (oz)
284	120
282	113
279	128

`1147 rows omitted…`

You're interested in constructing a test to determine whether there exists a relationship between the number of gestational days and the birth weight of the baby. This would suggest that the slope of the best fit line when using the gestational days to predict the birth weight is nonzero.

2.3 State the null and alternative hypotheses.

Null hypothesis:

Alternative hypothesis:

Next, we will bootstrap our sample and repeat the regression process to estimate the variability of the regression slope. Assume we already have a function `correlation(tbl, x, y)` that returns the correlation coefficient between two columns, `x` and `y`, in the table `tbl`. Additionally, we have already defined the following `slope` function.

```
def slope(table, x, y):
    r = correlation(table, x, y)
    return r * np.std(table.column(y))/np.std(table.column(x))
```

2.4 Complete the function below to make one bootstrapped sample of the `baby` table, and calculate the slope of the best fit line of that bootstrapped sample.

Hint: You can use the slope function defined above!

```
def one_slope():
    bootstrapped_baby_table = _____________________________________
    slope = _______________________________________________________
    return _______________________________________________________
```

2.5 Using the `one_slope` function defined in 2.4, populate the array slopes with 10,000 bootstrapped slopes from the `baby` table.

```
slopes = _____________________________________________________________

for i in _____________________:
    bootstrapped_slope = _____________________________________________
    slopes = _____________________________________________________________
```

2.6 Find the endpoints of the 98% confidence interval for our bootstrapped slopes.

```
lower_bound = _____________________________________________________________

upper_bound = _____________________________________________________________
```

2.7 Let's say we get the 98% confidence interval `(0.356, 0.585)`

a) What is the p-value cutoff associated with our level of confidence? Do we reject or fail to reject the null hypothesis at this cutoff value?

b) At a p-value cutoff of 5%, are we able to make conclusions about the null hypothesis? If so, do we reject or fail to reject the null hypothesis?

c) At a p-value cutoff of 1%, are we able to make conclusions about the null hypothesis? If so, do we reject or fail to reject the null hypothesis?

Worksheet 11: Classification and Final Review

1. Classification

Key Concepts

Definition

Classification is used to make predictions based on existing data. Some questions that we can answer with classification are:

- Is person A going to vote for a certain politician?
- Is a certain purchase an instance of credit card fraud?
- Do I have a certain disease?

Observations: existing individuals in a population that you have data on.

Attributes: characteristics of the individuals that you will build the classifier for. In Data 8, attributes are binary (yes or no, 1 or 0).

Population: A larger group of individuals, who you don't know the attributes for. A classifier is built in order to predict the attributes of those in the population.

Training and Testing Data

The reason why we've made a classifier is so that we can make predictions on new data from our **underlying population**. How do we know whether the classifier we've made actually makes accurate predictions?

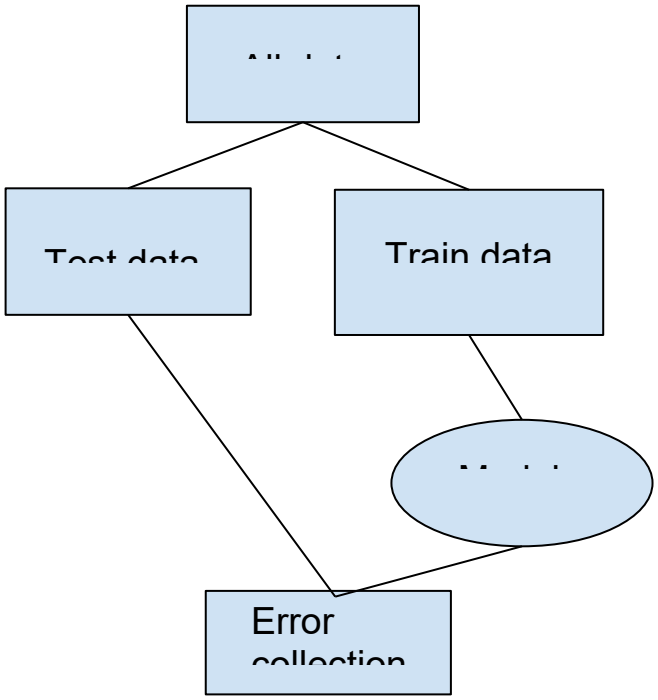
We can split our original dataset into two sets at random: **training data** and **testing data**, in order to validate the classifier's accuracy. A typical training and testing proportion split might be 80/20.

Practice Problems

1.1 Sue and Avery are deciding on which kind of nearest neighbors classifier they want to use. Avery says that using a larger number of neighbors will *always* result in more accurate predictions. Sue disagrees. Who is right, and why?

(Hint: Think about what happens when we have a data set with n points and we use an n-nearest neighbors classifier.)

2. Train/Test Split



2.1 In order to make the model as accurate as possible, should we use all of our data to train the model?

2.2 How should we split our data into training and testing sets? Why?

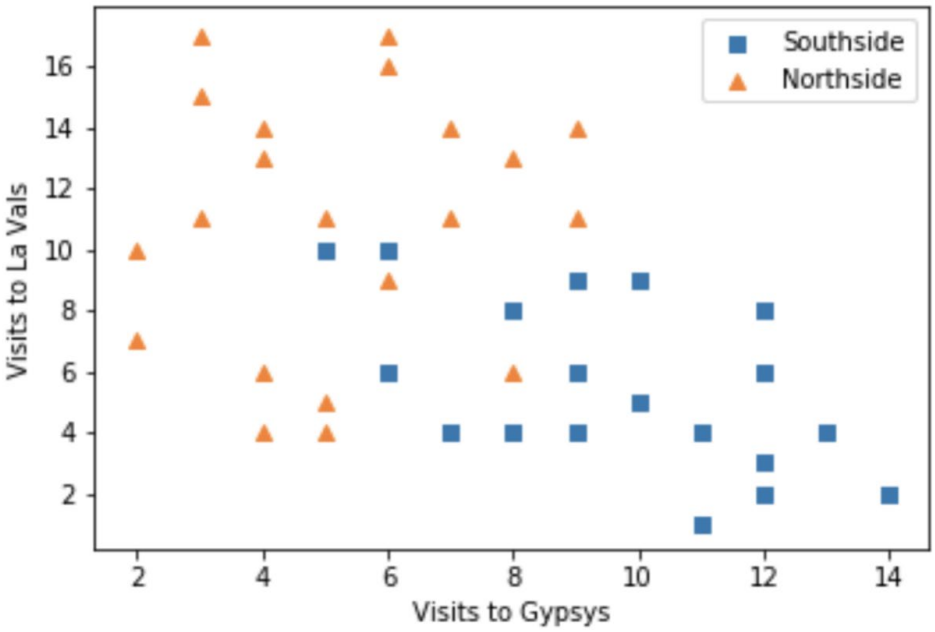
2.3 What would happen if we used 10% of the data for training, and 90% for testing? How much data do you think should be in the training set?

3. Decision Boundaries

A student is trying to build a classifier that classifies Berkeley students as residents of Northside or Southside. The student has a random sample of Berkeley students all of whom live on Northside or Southside. For each student she records whether the student lives on Northside or Southside, the number

of times the student went to La Val's (on Northside) in the last 6 months, and the number of times the student went to Gypsy's (on Southside) in the last 6 months.

3.1 Draw a decision boundary for a 5 nearest neighbor classifier on the scatter plot below.



1 Testing Chance Models

Roulette is a casino game in which a ball falls into one pocket in a spinning wheel, and players bet on the color of the pocket in which the ball falls. If the player correctly picks the color, they win. 18 of the 38 pockets in a roulette wheel are red. You play 30 games of roulette, bet on red every time, and win 20 of those games. You become suspicious about the fairness of this roulette game.

1.1 State a null and alternative hypothesis to see whether the roulette game is biased towards red.

Null Hypothesis:

Alternative Hypothesis:

1.2 With your alternative hypothesis in mind, choose a test statistic and calculate its observed value. Your test statistic should be large for data favoring the alternative hypothesis.

Test Statistic:

Observed Value:

1.3 Complete the function `prop_wins_in_30_games()`, which takes in no arguments and simulates playing the game 30 times and returns the proportion of wins when you guess red every time.

```
def prop_wins_in_30_games():
    proportions = _____________________________________________
    thirty_games = _____________________________________________
    prop_wins = _____________________________________________
    return _____________________________________________
```

1.4 Complete the code below to simulate an empirical distribution of the test statistic using 10000 iterations, storing the statistics in an array called `simulated_statistics`.

```
simulated_statistics = make_array()
for i in _____________________:
    prop_win = _____________________________________________
    simulated_statistics = _____________________________________________
```

1.5 Write a line of code to calculate the p-value.

1.6 Suppose you find a p-value of 0.0103. What do you conclude about the null hypothesis, at a p-value cutoff of 5%?

2 A/B Testing

We are examining the weights of a population of cats and dogs. You are given a random sample from this population, stored in the table `pets`, which has two columns. The first column `'Animal'` contains a string, either `'Cat'` or `'Dog'`. The second column `'Weight'` contains the weights of each of the animals in pounds as floats. You notice that the average weight of dogs in your sample is 2 pounds heavier than the average weight of cats in your sample.

2.1 State a null and alternative hypothesis to see if dogs weigh more than cats on average in the population.

Null Hypothesis:

Alternative Hypothesis:

2.2 With your alternative hypothesis in mind, choose a test statistic and calculate its observed value. Your test statistic should be large for data favoring the alternative hypothesis.

Test Statistic:

Observed Value:

2.3 Complete the function `one_shuffled_table_stat()` which takes in no arguments and returns one value of the test statistic.

```
def one_shuffed_table_stat():
    shuf_table = _____________________________________________
    shuf_weights = ____________________________________________
    shuf_tbl_with_weights = ___________________________________
    grouped_with_mean = _______________________________________
    dog_mean = ________________________________________________
    cat_mean = ________________________________________________
    return ____________________________________________________
```

2.4 Complete the below code to simulate an empirical distribution of 5000 test statistics under the assumptions of the null hypothesis.

```
diffs = make_array()
for i in np.arange(5000):
    diff = _____________________________________________________
    diffs = _____________________________________________________
```

2.5 Write a line of code to calculate your p-value.

```
p_value = _____________________________________________________
```

2.6 Suppose you find a p-value of 0.13. What do you conclude, at a p-value cutoff of 5%?

3 Regression Inference

You take a sample of Data 8 students and ask them about their daily consumption of coffee and their midterm exam scores. Assume you are given a table `students` with the columns `cups` and `score`. The column `cups` contains the daily consumption of coffee and score contains the midterm exam score for each student from the sample. You perform linear regression and find a slope of 0.13 points per cup of coffee.

3.1 State a null and alternative hypothesis to see whether this slope was due to randomness in your sample.

Null Hypothesis:
Alternative Hypothesis:

3.2 Write a function `slope` that takes in a table, `tbl`, and returns the slope of the least-squares line using the first column to predict values of the second column.

```
def slope(tbl):
    x = _____________________________________________________
    y = _____________________________________________________
    x_su = _____________________________________________________
    y_su = _____________________________________________________
    r = _____________________________________________________
    slope = _____________________________________________________
    return slope
```

3.3 Complete the code to generate 5000 bootstrap resample slopes and then calculate a 95% confidence interval for the slope. Assume the function `slope(tbl)` has been implemented correctly.

```
slopes = _____________________________________________________________
for i in _____________________:
    resample_slope = _____________________________________________
    slopes = _____________________________________________________

left_end = ____________________________________________________________
right_end = ___________________________________________________________
interval = make_array(left_end, right_end)
```

3.4 Suppose you find the confidence interval [0.02, 0.24]. What do you conclude about your hypotheses at a p-value cutoff of 5%? What about at a p-value cutoff of 10%? What about at a p-value cutoff of 1%?

3.5 Your friend who hasn't taken Data 8 looks at your result and asks you what the confidence interval means. Which one of the following is a correct response?

a) For 95% of students, there is a relationship between coffee consumption and midterm score.
b) There is a 95% probability that the true slope is between 0.02 and 0.24.
c) There is a 95% probability that our sampling process (and the code above) produces an interval that contains the true slope.

Appendix

Random Functions Guide

In Data 8, we have three methods of generating random values.

1. np.random.choice(array, sample_size=1)

   Returns an array of length sample_size consisting of items randomly sampled with replacement from the array. If you don't specify the sample_size argument, it returns just one randomly chosen item.

2. tbl.sample(n, with_replacement=False)

   This function takes a sample size n (an integer), and an optional argument with_replacement, which is a boolean. This function returns a new table where n rows are randomly sampled from the original table; by default, n=tbl.num_rows. Default is with_replacement=True. For sampling without replacement, use argument with_replacement=False.

3. sample_proportions(sample_size, model_proportions)

   sample_size should be an integer, model_proportions an array of probabilities that sum up to 1. The function samples sample_size objects from the distribution specified by model_proportions. It returns an array with the same size as model_proportions. Each item in the array corresponds to the proportion of times that item was sampled out of the sample_size times. Both the input model_proportions and output sum to 1.

Each of these functions can be used interchangeably to sample with replacement. To sample without replacement, we use tbl.sample(n, with_replacement=False).

Let's see how we can use these functions interchangeably.

To simulate 1000 coin flips of a fair coin, we could write:

1. np.random.choice(make_array("Heads", "Tails"), 1000)
   This will return an array of 1000 string values that are either "Heads" or "Tails".
2. Table().with_column("Outcome", make_array("Heads", "Tails")).sample(1000)
   This will return a table with 1000 rows, and a column "Outcome". Each row will have either the string "Heads" or the string "Tails" in it.
3. sample_proportions(1000, make_array(0.5, 0.5))
   This will return an array with two elements in it, the first item will be the simulated proportion of heads in 1000 tosses, the second item will be the simulated proportion of tails in those 1000 tosses.

To count the number of times the face with 3 dots appears in 500 rolls of a fair dice we could write any of the following:

1. rolls = np.random.choice(np.arange(1, 7), 500)
   num_3 = np.count_nonzero(rolls == 3)

2. dice = Table().with_column("Face", np.arange(1, 7))
   rolls = dice.sample(500)
   num_3 = rolls.where("Face", are.equal_to(3)).num_rows

3. model_proportions = make_array(1/6, 1/6, 1/6, 1/6, 1/6, 1/6)
   rolls = sample_proportions(500, model_proportions)
   num_3 = rolls.item(2) * 500

When should we use a specific function?

Because each of these functions are interchangeable there are no set rules for which function to use. That being said, we can save ourselves some time by following these tips:

1. If we have some model_proportions specified in our problem, sample_proportions will probably be the simplest function to use!
2. If we are dealing with resampling values or our data is stored in a table already then tbl.sample will probably be the most convenient function to use.
3. If we are choosing from an array of values, np.random.choice is often the easiest function to implement, or we can store the values in a table and then use tbl.sample.
4. When shuffling the labels for A/B testing we should always use tbl.sample(with_replacement=False) because this is the only way we can sample without replacement.

These are only general tips--if you want more specific examples of when you might use each function, check out the textbook! For example, section 10.2 has an example of using tbl.sample to sample from a table of flights, and section 11.2 has an illustrative example on when to use sample_proportions.

Reference Guides

Data 8 Final Reference Guide — Page 1

Statements

```
more_than_1 = 2 + 3
```
Name — Any expression

- Statements don't have a value; they perform an action
- An assignment statement changes the meaning of the name to the left of the = symbol
- The name is bound to a value (not an equation).

Comparisons

- < and > mean what you expect (less than, greater than)
- <= means "less than or equal"; likewise for >=
- == means "equal"; != means "not equal"
- Comparing strings compares their alphabetical order

Arrays - sequences of the same type that can be manipulated

- Arithmetic and comparisons are applied to each element of an array individually
  - `make_array(1,2,3) ** 2 # array([1, 4, 9])`
- Elementwise operations can be done on arrays of the same size
  - `make_array(3,2) * make_array(5,4) # array([15,8])`

Defining a Function

```
def function_name(arg1, arg2, ...):
    # Body can contain anything inside of it
    return # a value (the output of the function call)
```

Defining a Function with no arguments

```
def function_name():
    # Body can contain anything inside of it
    return # a value (the output of the function call)
```

- Functions with no arguments can be called by `function_name()`

For Statements

```
total = 0
for i in np.arange(12):
    total = total + i
```

- The body is executed **for** every item in a sequence
- The body of the statement can have multiple lines
- The body should do something: assign, sample, print, etc.

Conditional Statements

```
if <if expression>:
    <if body>
elif <elif expression 0>:
    <elif body 0>
elif <elif expression 1>:
    <elif body 1>
...
else:
    <else body>
```

Total Variation Distance

Total variation distance is a statistic that represents the difference between two distributions

```
TVD = 0.5*(np.sum(np.abs(dist1-dist2)))
```

Operations: addition `2+3=5`; subtraction `4-2=2`; division `9/2=4.5` multiplication `2*3=6`; division remainder `11%3=2`; exponentiation `2**3=8`

Data Types: string "hello"; **boolean** `True, False`; **int** `1, -5`; **float** `- 2.3, -52.52, 7.9, 8.0`

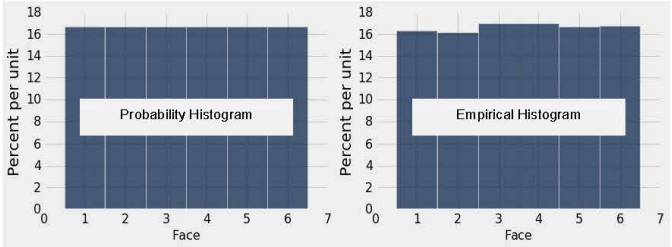
`Table.where` predicates: Any of these predicates can be negated by adding "not_" in front of them, e.g. `are.not_equal_to(x)`

- `are.equal_to(x) # val == x`
- `are.above(x) # val > x`
- `are.above_or_equal_to(x) # val >= x`
- `are.below(x) # val < x`
- `are.between(x, y) # x <= val < y`
- `are.containing(s) # contains the string s`

A **histogram** has a few defining properties:

- The bins are continuous (though some might be empty) and are drawn to scale
- The **area** of each bar is equal to the percent of entries in the bin
- The total area is 100%

- The histogram on the left represents the theoretical probabilities in the distribution of the face that appears on one roll of a fair die
- The histogram on the right represents the observed distribution of the faces after rolling the die many times
- If we keep rolling, the right hand histogram is likely to look more like the one on the left



Calculating Probabilities

Complement Rule: P(event does not happen) = 1 - P(event happens)

Multiplication Rule: P(two events both happen) =
P(one happens) * P(the other happens, given that the first happened)

Addition Rule: If an event can happen in ONLY one of two ways:
P(event happens) =
P(first way it can happen) + P(second way it can happen)

Bayes' Rule: P(event A happened given event B happened) =
P(both event A and event B happened) / P(event B happened)

For Bayes' rule, if the probabilities are displayed on a tree diagram, the denominator is the chance of the branches in which B happens, and the numerator is the chance of the branches in which both A and B happen.

Simulating a Statistic:

- Create an empty array in which to collect the simulated values
- For each repetition of the process
  - Simulate one value of the statistic
  - Append this value to the collection array
- At the end, all simulated values will be in the collection array

Data 8 Final Reference Guide — Page 2

In the examples in the left column, `np` refers to the NumPy module, as usual. Everything else is a function, a method, an example of an argument to a function or method, or an example of an object we might call the method on. For example, `tbl` refers to a table, `array` refers to an array, and `num` refers to a number. `array.item(0)` is an example call for the method `item`, and in that example, `array` is the name previously given to some array.

`max(array); min(array)`	Maximum or minimum of an array
`sum(array)`	Sum of all elements in an array; The sum of an array of boolean values is the number of values that are True
`len(array)`	Length (num elements) in an array
`round(num); np.round(array)`	The nearest integer to a single number or each number in an array
`abs(num); np.abs(array)`	The absolute value of a single number or each number in an array
`np.average(array), np.mean(array)`	The average of the values in an array

`np.arange(start, stop, step)`
`np.arange(start, stop)`
`np.arange(stop)`
	An array of numbers starting with `start`, going up in increments of `step`, and going up to but excluding `stop`. When `start` and/or `step` are left out, default values are used in their places. Default `step` is 1; default `start` is 0.

`array.item(index)`	The item in the array at some index. `array.item(0)` is the first item of `array`.
`np.append(array, item)`	A copy of the array with item appended to the end. If item is another array, all of its elements are appended.
`np.exp(array)`	Calculate the exponential fo all the elements in the array.
`np.random.choice(array)` `np.random.choice(array, n)`	An item selected at random from an array. If `n` is specified, an array of n items selected at random with replacement is returned. Default `n` is 1.
`np.ones(n)`	An array of length `n` which consists of all ones.
`np.diff(array)`	An array of length `len(array)-1` which contains the difference between adjacent elements.
`np.count_nonzero(array)`	An integer corresponding to the number of non-zero (or `True`) elements in an array.
`sample_proportions(sample_size, model_proportions)`	An array of proportions that add up to 1. The result of sampling `sample_size` elements from a distribution specified by `model_proportions`, and keeping track of the proportion of each element sampled.
`Table()`	An empty table.
`Table.read_table(filename)`	A table with data from a file.
`tbl.num_rows`	The number of rows in a table.
`tbl.num_columns`	The number of columns in a table.
`tbl.labels`	A list of the column labels of a table.
`tbl.with_column(name, values)` `tbl.with_columns(n1, v1, n2, v2...)`	A table with an additional or replaced column or columns. `name` is a string for the name of a column, `values` is an array.
`tbl.column(column_name_or_index)`	An array containing the values of a column
`tbl.select(col1, col2, ...)`	A table with only the selected columns. (Each argument is the label of a column, or a column index.)
`tbl.drop(col1, col2, ...)`	A table without the dropped columns. (Each argument is the label of a column, or a column index.)
`tbl.relabeled(old_label, new_label)`	A new table with a label changed.
`tbl.take(row_index)` `tbl.take(row_indices)`	A table with only the row(s) at the given index or multiple indices. `row_indices` must be an array of indices.
`tbl.exclude(row_index)` `tbl.exclude(row_indices)`	A table without the row(s) at the index or multiple indices. `row_indices` must be an array of indices.
`tbl.sort(column_name_or_index)`	A table of rows sorted according to the values in a column (specified by name/index). Default order is ascending. For descending order, use argument `descending=True`. For unique values, use `distinct=True`.
`tbl.where(column, predicate)`	A table of the rows for which the column satisfies some predicate. See "`Table.where predicates`" on Page 1.
`tbl.apply(function, column_or_columns)`	An array of results when a `function` is applied to each item in a `column`.
`tbl.group(column_or_columns)`	A table with the counts of rows grouped by unique values or combinations of values in a column or columns.
`tbl.group(column_or_columns, func)`	A table that groups rows by unique values or combinations of values in a column or columns. The other values are aggregated by `func`. All column names (except the one(s) we group by) will now be `original_name func`. If a column is named 'price', and we group using the `min` function, our new column name will be 'price min'.
`tblA.join(colA, tblB, colB)` `tblA.join(colA, tblB)`	A table with the columns of `tblA` and `tblB`, containing rows for all values of a column that appear in both tables. Default value of `colB` is `colA`. `colA` is a string specifying a column name, as is `colB`.
`tbl.pivot(col1, col2)` `tbl.pivot(col1, col2, vals, collect)`	A pivot table where each unique value in `col1` has its own column and each unique value in `col2` has its own row. The cells of the grid contain row counts (two arguments) or the values from a third column, aggregated by the collect function (four arguments) .
`tbl.sample(n)` `tbl.sample(n, with_replacement)`	A new table where `n` rows are randomly sampled from the original table. Default is with replacement. For sampling without replacement, use argument `with_replacement=False`. If sample size `n` is not specified, the default is the number of rows in the original table.
`tbl.scatter(x_column, y_column)`	Draws a scatter plot consisting of one point for each row of the table.
`tbl.barh(categories)` `tbl.barh(categories, values)`	Displays a bar chart with bars for each category in a column, with length proportional to the corresponding frequency. If `values` is not specified, overlaid bar charts of all the remaining columns are drawn.
`tbl.bin(column, bins)`	A table of how many values in a column fall into each bin. Bins include lower bounds & exclude upper bounds.
`tbl.hist(column, unit, bins, group)`	Displays a histogram of the values in a column. `unit` and `bins` are optional arguments, used to label the axes and group the values into intervals (bins), respectively. Bins include lower bounds & exclude upper bounds. If `group` is specified, the rows are grouped by the values in the column, and histograms for all the groups are overlaid.

- **P-Value**: The chance, **under the null hypothesis**, that the test statistic comes out equal to the one in the sample, or more in the direction of the alternative:
  - If the p-value is small and the null is true, something very unlikely has happened.
  - Conclude that the data support the alternative hypothesis more than they support the null.

- Even if the null is true, your random sample might indicate the alternative, just by chance
- The **cutoff** for P is the chance that your test makes the wrong conclusion when the null hypothesis is true
- Using a small cutoff limits the probability of this kind of error

---

A/B test for comparing two samples

- **Example**: Among babies born at some hospital, is there an association between birth weight and whether the mother smokes?
- **Null hypothesis**: The distribution of birth weights is the same for babies with smoking mothers and non-smoking mothers.
- **Inferential Idea:** If maternal smoking and birth weight were not associated, then we could simulate new samples by replacing each baby's birth weight by a randomly picked value from among all the birth weights.
- **Simulating the test statistic under the null:**
  - Permute (shuffle) the outcome column many times. Each time:
    - Create a shuffled table that pairs each individual with a random outcome.
    - Compute a sampled test statistic that compares the two groups, such as the difference in mean birth weights.

The 80th percentile is the value in a set that is at least as large as 80% of the elements in the set

For `s = [1, 7, 3, 9, 5]`, `percentile(80, s)` is 7

The 80th percentile is ordered element 4: `(80/100) * 5`

For a percentile that does not exactly correspond to an element, take the next greater element instead

`percentile(10, s)` is 1 `percentile(20, s)` is 1
`percentile(21, s)` is 3 `percentile(40, s)` is 3

- `minimize` must take in a function whose arguments are numerical, and returns an array of those numerical arguments
- If the function `rmse(a, b)` returns the root mean squared error of estimation using the line "estimate = a*x* + b",
  - then `minimize(rmse)` returns array [a_0, b_0]
  - a_0 is the slope and b_0 the intercept of the line that minimizes the rmse among lines with arbitrary slope a and arbitrary intercept b (that is, among all lines)

Population (fixed) → Sample (random) → Statistic (random)

A 95% **Confidence Interval** is an interval constructed so that it will contain the true population parameter for approximately 95% of samples

For a particular sample, the generated interval either contains the true parameter or it doesn't; the process works 95% of the time

Bootstrap: When we wish we could sample again from the population, instead sample from the *original large random sample the same number of times as there are data-points in the sample*

Using a confidence interval to test a hypothesis about a numerical parameter:

- Null hypothesis: **Population parameter = *x***
- Alternative hypothesis: **Population parameter ≠ *x***
- Cutoff for P-value: *p*%
- Method:
  - Construct a (100-*p*)% confidence interval for the population parameter
  - If *x* is not in the interval, reject the null
  - If *x* is in the interval, fail to reject the null

The Central Limit Theorem (CLT)

If the sample is large, and drawn at random with replacement,

Then, *regardless of the distribution of the population,*

the probability distribution of the sample average (or sample sum) is roughly bell-shaped

- Fix a large sample size
- Draw all possible random samples of that size
- Compute the mean of each sample
- You'll end up with a lot of means
- The distribution of those is the *probability distribution of the sample mean*
- It's roughly normal, centered at the population mean
- The SD of this distribution is the (population SD) /√sample size

---

Choosing sample size so that the 95% confidence interval is small

- CLT says the distribution of a sample proportion is roughly normal, centered at the true population proportion
- **95% confidence interval:**
  - Sample proportion ± 2 SDs of the sample proportion
- **CI Width** = 4 SDs of the sample proportion

  = 4 x (SD of 0/1 population)/√sample size
- The SD of a 0/1 population is less than or equal to 0.5

Expression	Description
`percentile(n, arr)`	Returns the n-th percentile of array `arr`
`np.std(arr)`	Return the standard deviation of an array `arr` of numbers
`minimize(fn)`	Return an array of arguments that minimize the function `fn`
`tbl1.append(tbl2)` `tbl1.append(row)`	Append a row or all rows of `tbl2`, mutating `tbl1`. Appended object and `tbl1` must have identical columns.
`table.rows`	All rows of a table; used in `for row in table.rows:`
`table.row(i)`	Return the row of a table at index *i*
`row.item(j)`	Returns item `j` from some row

Mean (or average): Balance point of the histogram

Standard deviation (SD) =

root	mean	square of	deviations from	average
5	4	3	2	1

Measures roughly how far off the values are from average

Most values are within the range "average ± *z* SDs"

- *z* measures "how many SDs above average"
- If *z* is negative, the value is below average
- *z* is a value in **standard units**
- Chebyshev: At most $1/z^2$ are z or more SDs from the mean
- Almost all standard unit values are in the range (-5, 5)
- Convert a value to standard units: (value - average) / SD
- *z* * SD + average is the original value

Percent in Range	All Distributions	Normal Distribution
average ± 1 SD	at least 0%	about 68%
average ± 2 SDs	at least 75%	about 95%
average ± 3 SDs	at least 88.888...%	about 99.73%

Correlation Coefficient (*r*) =

average of	product of	x in standard units	and	y in standard units

Measures how clustered the scatter is around a straight line

- $-1 \le r \le 1$; r = 1 (or -1) if the scatter is a perfect straight line
- *r* is a pure number, with no units

Regression for y and x in standard units: $y_{predicted,su} = r * x_{su}$

The regression line minimizes the root mean square error among all lines used to predict y from x.
The slope and intercept found by linear regression are unique.
Fitted value: height of the regression line at some x: a*x + b
Residual: difference between y and regression line height at x

$$y_{predicted} = slope * x + intercept$$

$$\text{slope of the regression line} = r \cdot \frac{\text{SD of } y}{\text{SD of } x}$$

$$\text{intercept of the regression line} = \text{mean of } y - \text{slope} * \text{mean of } x$$

Properties of fitted values, residuals, and the correlation r:

- mean of fitted values = mean of y
- SD of fitted values = |r| * (SD of y)
- mean of residuals = 0
- SD of residuals = $\sqrt{1 - r^2}$ * (SD of y)

The following functions were defined in lecture, but will **not** be available for use during the final exam. If you would like to use one of the functions, you must define it yourself.

- `standard_units`
- `correlation`
- `slope`
- `intercept`
- `fitted_values`
- `residuals`
- `prediction_at`

- **Regression Model**: y is a linear function of x + normal "noise"
- The errors are randomly sampled from a normal distribution that has mean 0
- Under this model, residual plot looks like a formless cloud

Prediction Intervals (assuming the regression model)

- Creating an interval of predictions of the true value of *y* based on a specified value of *x*
- Steps for creating an approximate 95% prediction interval:
  - Bootstrap your original sample
  - Calculate the slope and intercept of the regression line based on the new sample
  - Calculate slope * x + intercept, for the given *x*
  - Repeat the above steps many times and keep track of all of your fitted values
  - Create the prediction interval by taking the middle 95% of all the fitted values

Distance between two points

- Two numerical attributes *x* and *y*: $D = \sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2}$.
- Three numerical attributes *x*, *y*, and *z*:

$$D = \sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2 + (z_0 - z_1)^2}$$

***k*-Nearest Neighbors Classifier**

Choose *k* to be odd. To find the *k* nearest neighbors of an example:

- Find the distance between the example and each example in the training set
- Augment the training data table with a column containing all the distances
- Sort the augmented table in increasing order of the distances
- Take the top *k* rows of the sorted table

To classify an example into one of two classes:

- Find its *k* nearest neighbors
- Take a majority vote of the *k* nearest neighbors to see which of the two classes appears more often
- Assign the example the class that wins the majority vote

Accuracy of a classifier: The proportion of examples in the data set that are classified correctly

Lab, Homework, and Lecture Notebooks