Homework 8: Confidence Intervals

Reading:

- Estimation (https://www.inferentialthinking.com/chapters/13/estimation.html)

Please complete this notebook by filling in the cells provided. Before you begin, execute the following cell to load the provided tests. Each time you start your server, you will need to execute this cell again to load the tests.

Homework 8 is due **Thursday, 4/2 at 11:59pm**. You will receive an early submission bonus point if you turn in your final submission by Wednesday, 4/1 at 11:59pm. Start early so that you can come to office hours if you're stuck. Check the website for the office hours schedule. Late work will not be accepted as per the policies (http://data8.org/sp20/policies.html) of this course.

Directly sharing answers is not okay, but discussing problems with the course staff or with other students is encouraged. Refer to the policies page to learn more about how to learn cooperatively.

For all problems that you must write our explanations and sentences for, you **must** provide your answer in the designated space. Moreover, throughout this homework and all future ones, please be sure to not re-assign variables throughout the notebook! For example, if you use `max_temperature` in your answer to one question, do not reassign it later on.

In [1]:
```
# Don't change this cell; just run it.

import numpy as np
from datascience import *

# These lines do some fancy plotting magic.
import matplotlib
%matplotlib inline
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')
import warnings
warnings.simplefilter('ignore', FutureWarning)

from client.api.notebook import Notebook
ok = Notebook('hw08.ok')
```

```
=====================================================================
Assignment: Homework 8: Confidence Intervals
OK, version v1.14.19
=====================================================================
---------------------------------------------------------------------------
LoadingException                          Traceback (most recent call last)
<ipython-input-1-4ffb8436620b> in <module>
     13 
     14 from client.api.notebook import Notebook
---> 15 ok = Notebook('hw08.ok')

/opt/anaconda3/lib/python3.7/site-packages/client/api/notebook.py in __init__(self, filepath, cmd_args, debug, mode)
     13         ok_logger = logging.getLogger('client')   # Get top-level ok logger
     14         ok_logger.setLevel(logging.DEBUG if debug else logging.ERROR)
---> 15         self.assignment = load_assignment(filepath, cmd_args)
     16         # Attempt a login with enviornment based tokens
     17         login_with_env(self.assignment)

/opt/anaconda3/lib/python3.7/site-packages/client/api/assignment.py in load_assignment(filepath, cmd_args)
     22     if cmd_args is None:
     23         cmd_args = Settings()
---> 24     return Assignment(cmd_args, **config)
     25 
     26 def _get_config(config):

/opt/anaconda3/lib/python3.7/site-packages/client/sources/common/core.py in __call__(cls, *args, **kargs)
    185                 raise ex.SerializeException('__init__() missing expected '
    186                                 'argument {}'.format(attr))
--> 187         obj.post_instantiation()
    188         return obj
    189 

/opt/anaconda3/lib/python3.7/site-packages/client/api/assignment.py in post_instantiation(self)
    151     def post_instantiation(self):
    152         self._print_header()
--> 153         self._load_tests()
    154         self._load_protocols()
    155         self.specified_tests = self._resolve_specified_tests(

/opt/anaconda3/lib/python3.7/site-packages/client/api/assignment.py in _load_tests(self)
    205 
    206         if not self.test_map:
--> 207             raise ex.LoadingException('No tests loaded')
    208 
    209     def dump_tests(self):

LoadingException: No tests loaded
```

1. Thai Restaurants

Ben and Frank are trying see what the best Thai restaurant in Berkeley is. They survey 1500 UC Berkeley students selected uniformly at random, and ask each student what Thai restaurant is the best (*Note: this data is fabricated for the purposes of this homework*). The choices of Thai restaurant are Lucky House, Imm Thai, Thai Temple, and Thai Basil. After compiling the results, Ben and Frank release the following percentages from their sample:

Thai Restaurant	Percentage
Lucky House	8%
Imm Thai	52%
Thai Temple	25%
Thai Basil	15%

These percentages represent a uniform random sample of the population of UC Berkeley students. We will attempt to estimate the corresponding *parameters*, or the percentage of the votes that each restaurant will receive from the entire population (the entire population is all UC Berkeley students). We will use confidence intervals to compute a range of values that reflects the uncertainty of our estimates.

The table `votes` contains the results of the survey.

In [2]:
```
# Just run this cell
votes = Table.read_table('votes.csv').sample(with_replacement = False)
votes
```

Out[2]:

Vote
Imm Thai
Imm Thai
Imm Thai
Thai Basil
Imm Thai
Thai Temple
Thai Temple
Imm Thai
Imm Thai
Thai Basil

... (1490 rows omitted)

Question 1. Complete the function `one_resampled_percentage` below. It should return Imm Thai's **percentage** of votes after simulating one bootstrap sample of `tbl` .

Note: `tbl` will always be in the same format as `votes` .

```
BEGIN QUESTION
name: q1_1
manual: false
```

In [3]:
```
def one_resampled_percentage(tbl):
    # BEGIN SOLUTION
    bootstrap = tbl.sample()
    single_percentage = (np.count_nonzero(bootstrap.column('Vote') == 'I
mm Thai') / votes.num_rows) * 100
    return single_percentage
    # END SOLUTION

one_resampled_percentage(votes)
```

Out[3]:
```
51.266666666666667
```

In [4]:
```
# TEST
type(one_resampled_percentage(votes)) in set([float, np.float64])
```

Out[4]:
```
True
```

In [5]:
```
# TEST
35 <= one_resampled_percentage(votes) <= 65
```

Out[5]:
```
True
```

In [6]:
```
# HIDDEN TEST
np.random.seed(123)
one_resampled_percentage(votes)
```

Out[6]:
```
50.46666666666667
```

Question 2. Complete the `percentages_in_resamples` function such that it returns an array of 2500 bootstrapped estimates of the percentage of voters who will vote for Imm Thai. You should use the `one_resampled_percentage` function you wrote above.

Note: There are no public tests for this question, the autograder cell below will return 0.0% passed.

```
BEGIN QUESTION
name: q1_2
manual: false
```

In [7]:
```
def percentages_in_resamples():
    percentage_imm = make_array()
    # BEGIN SOLUTION
    for i in np.arange(2500):
        single_percentage = one_resampled_percentage(votes)
        percentage_imm = np.append(percentage_imm, single_percentage)
    return percentage_imm
    # END SOLUTION
```

In [8]:
```
# HIDDEN TEST
len(percentages_in_resamples())
```

Out[8]:
```
2500
```

In [9]:
```
# HIDDEN TEST
np.random.seed(123)
percentages_in_resamples().item(0)
```

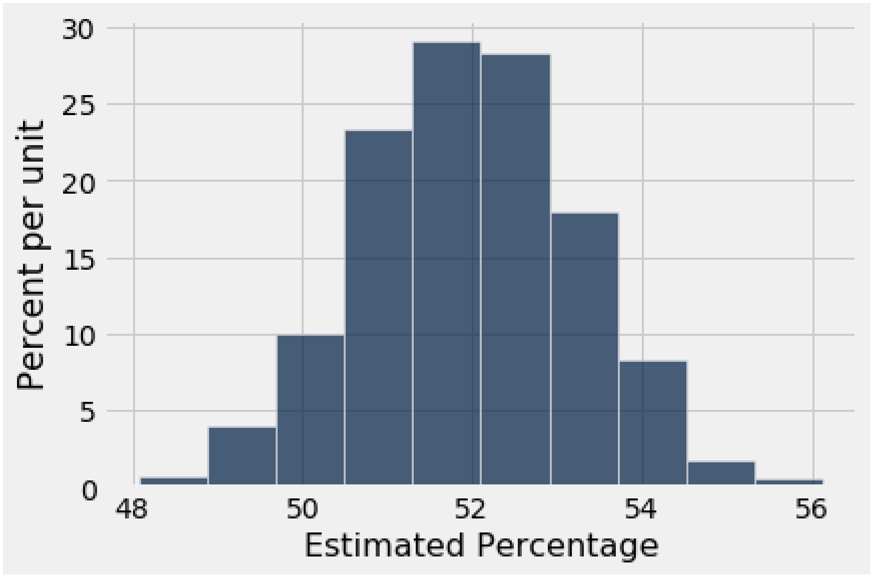
Out[9]:
```
50.46666666666667
```

In [10]:
```
# HIDDEN TEST
np.random.seed(123)
percentages_in_resamples().item(10)
```

Out[10]:
```
51.4
```

In the following cell, we run the function you just defined, `percentages_in_resamples`, and create a histogram of the calculated statistic for the 2,500 bootstrap estimates of the percentage of voters who voted for Imm Thai. Based on what the original Thai restaurant percentages were, does the graph seem reasonable? Talk to a friend or ask a TA if you are unsure!

In [11]:
```
resampled_percentages = percentages_in_resamples()
Table().with_column('Estimated Percentage', resampled_percentages).hist(
"Estimated Percentage")
```



Question 3. Using the array `resampled_percentages`, find the values at the two edges of the middle 95% of the bootstrapped percentage estimates. (Compute the lower and upper ends of the interval, named `imm_lower_bound` and `imm_upper_bound`, respectively.)

```
BEGIN QUESTION
name: q1_3
manual: false
```

In [12]:
```
imm_lower_bound = percentile(2.5, resampled_percentages) # SOLUTION
imm_upper_bound = percentile(97.5, resampled_percentages) # SOLUTION
print("Bootstrapped 95% confidence interval for the percentage of Imm Th
ai voters in the population: [{:f}, {:f}]".format(imm_lower_bound, imm_u
pper_bound))
```

```
Bootstrapped 95% confidence interval for the percentage of Imm Thai vot
ers in the population: [49.466667, 54.400000]
```

In [13]:
```
# TEST
40 <= imm_lower_bound <= imm_upper_bound <= 60
```

Out[13]:
```
True
```

In [14]:
```
# HIDDEN TEST
all([imm_lower_bound == percentile(2.5, resampled_percentages), imm_uppe
r_bound == percentile(97.5, resampled_percentages)])
```

Out[14]:
```
True
```

Question 4. The survey results seem to indicate that Imm Thai is beating all the other Thai restaurants combined among voters. We would like to use confidence intervals to determine a range of likely values for Imm Thai's true lead over all the other restaurants combined. The calculation for Imm Thai's lead over Lucky House, Thai Temple, and Thai Basil combined is:

$$\text{Imm Thai's \% of the vote} - (\text{Lucky House's \% of the vote} + \text{Thai Temple's \% of the vote} + \text{Thai Basil's \%}$$

Define the function `one_resampled_difference` that returns **exactly one value** of Imm Thai's percentage lead over Lucky House, Thai Temple, and Thai Basil combined from one bootstrap sample of `tbl`.

```
BEGIN QUESTION
name: q1_4
manual: false
```

In [15]:
```
def one_resampled_difference(tbl):
    bootstrap = tbl.sample() #SOLUTION
    imm_percentage = (np.count_nonzero(bootstrap.column('Vote') == 'Imm
 Thai') / tbl.num_rows) * 100 #SOLUTION
    lh_percentage = (np.count_nonzero(bootstrap.column('Vote') == 'Lucky
House') / tbl.num_rows) * 100 #SOLUTION
    tt_percentage = (np.count_nonzero(bootstrap.column('Vote') == 'Thai
 Temple') / tbl.num_rows) * 100 #SOLUTION
    tb_percentage = (np.count_nonzero(bootstrap.column('Vote') == 'Thai
 Basil') / tbl.num_rows) * 100 #SOLUTION
    return imm_percentage - (lh_percentage + tt_percentage + tb_percenta
ge) #SOLUTION
```

In [16]:
```
# TEST
type(one_resampled_difference(votes)) in set([float, np.float64])
```

Out[16]:
```
True
```

In [17]:
```
# HIDDEN TEST
np.random.seed(123)
-6 <= float(one_resampled_difference(votes)) <= 15
```

Out[17]:
```
True
```

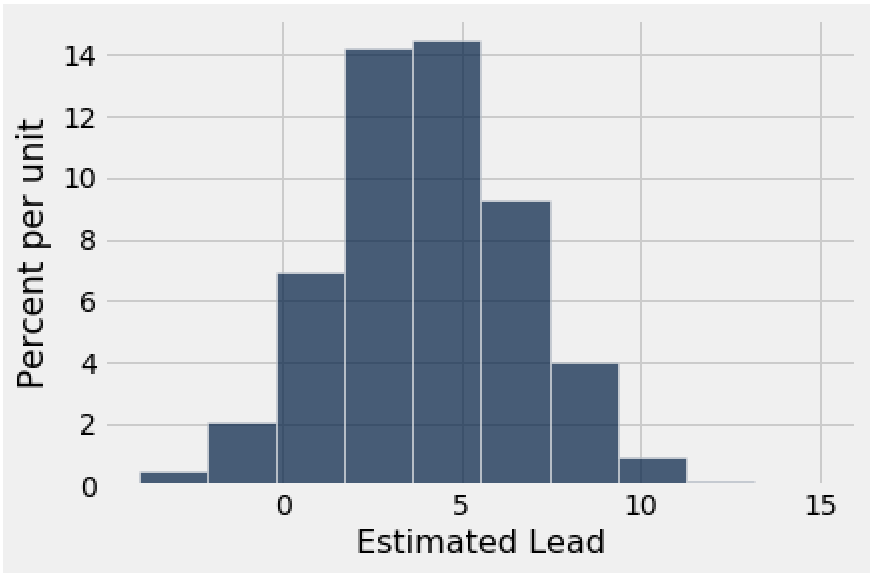
Question 5. Write a function called `leads_in_resamples` that finds 2,500 bootstrapped estimates (the result of calling `one_resampled_difference`) of Imm Thai's lead over Lucky House, Thai Temple, and Thai Basil combined. Plot a histogram of the resulting samples.

Note: Imm Thai's lead can be negative.

```
BEGIN QUESTION
name: q1_5
manual: true
```

In [18]:
```
def leads_in_resamples():
    # BEGIN SOLUTION
    leads = make_array()
    for i in np.arange(2500):
        bootstrap_lead = one_resampled_difference(votes)
        leads = np.append(leads, bootstrap_lead)
    return leads
    # END SOLUTION

sampled_leads = leads_in_resamples()
Table().with_column('Estimated Lead', sampled_leads).hist("Estimated Lea
d")
```



Question 6. Use the simulated data from Question 5 to compute an approximate 95% confidence interval for Imm Thai's true lead over Lucky House, Thai Temple, and Thai Basil combined.

```
BEGIN QUESTION
name: q1_6
manual: false
```

In [19]:
```
diff_lower_bound = percentile(2.5, sampled_leads) # SOLUTION
diff_upper_bound = percentile(97.5, sampled_leads) # SOLUTION
print("Bootstrapped 95% confidence interval for Imm Thai's true lead ove
r Lucky House, Thai Temple, and Thai Basil combined: [{:f}, {:f}]".forma
t(diff_lower_bound, diff_upper_bound))
```

```
Bootstrapped 95% confidence interval for Imm Thai's true lead over Luck
y House, Thai Temple, and Thai Basil combined: [-1.066667, 9.200000]
```

In [20]:
```
# TEST
-5 <= diff_lower_bound <= diff_upper_bound <= 12
```

Out[20]:
```
True
```

```
In [21]: # HIDDEN TEST
         -1.25 <= diff_lower_bound <= diff_upper_bound <= 11

Out[21]: True
```

```
In [22]: # HIDDEN TEST
         all([diff_lower_bound == percentile(2.5, sampled_leads), diff_upper_boun
         d == percentile(97.5, sampled_leads)])

Out[22]: True
```

2. Interpreting Confidence Intervals

The staff computed the following 95% confidence interval for the percentage of Imm Thai voters:

$$[49.40, 54.47]$$

(Your answer may have been a bit different; that doesn't mean it was wrong!)

Question 1

Can we say there is a 95% probability that the interval [49.40, 54.47] contains the true percentage of the population that votes for Imm Thai as the best Berkeley Thai restaurant? Answer "yes" or "no" and explain your reasoning.

Note: ambiguous answers using language like "sometimes" or "maybe" will not receive credit.

```
BEGIN QUESTION
name: q2_1
manual: true
```

SOLUTION: No, the true proportion is some value x. Our observed interval $[49.40, 54.47]$ has already been fixed, so x is either in the interval or it is not.

Question 2

The staff also created 70%, 90%, and 99% confidence intervals from the same sample, but we forgot to label which confidence interval represented which percentages! Match each confidence level (70%, 90%, 99%) with its corresponding interval in the cell below (e.g. __ % CI: [49.87, 54.0] → replace the blank with one of the three confidence levels). **Then**, explain your thought process and how you came up with your answers.

The intervals are below:

- [49.87, 54.00]
- [50.67, 53.27]
- [48.80, 55.40]

```
BEGIN QUESTION
name: q2_2
manual: true
```

SOLUTION:

70% CI: [50.67, 53.27]

90% CI: [49.87, 54.00]

99% CI: [48.80, 55.40]

We compute these intervals by taking the middle X% of a bunch of bootstrap statistics. As the confidence level increases, we are including more and more of the statistics, so the interval widens. Intuitively, we might be very confident that the population parameter is within in some giant interval, but only moderately confident that it's within some smaller interval.

Question 3

Suppose we produced 5,000 new samples (each one a uniform random sample of 1,500 voters/students) from the population and created a 95% confidence interval from each one. Roughly how many of those 5,000 intervals do you expect will actually contain the true percentage of the population?

Assign your answer to `true_percentage_intervals` .

```
BEGIN QUESTION
name: q2_3
manual: false
```

```
In [23]: true_percentage_intervals = 4750 # SOLUTION
In [24]: # TEST
         1000 <= true_percentage_intervals <= 10000
Out[24]: True
```

```
In [25]: # HIDDEN TEST
         true_percentage_intervals == 4750
Out[25]: True
```

Recall the second bootstrap confidence interval you created, which estimated Imm Thai's lead over Lucky House, Thai Temple, and Thai Basil combined. Among voters in the sample, Imm Thai's lead was 4%. The staff's 95% confidence interval for the true lead (in the population of all voters) was

$$[-0.80, 8.80]$$

Suppose we are interested in testing a simple yes-or-no question:

> "Is the percentage of votes for Imm Thai tied with the percentage of votes for Lucky House, Thai Temple, and Thai Basil combined?"

Our null hypothesis is that the percentages are equal, or equivalently, that Imm Thai's lead is exactly 0. Our alternative hypothesis is that Imm Thai's lead is not equal to 0. In the questions below, don't compute any confidence interval yourself - use only the staff's 95% confidence interval.

Question 4

Say we use a 5% P-value cutoff. Do we reject the null, fail to reject the null, or are we unable to tell using our staff confidence interval?

Assign `restaurants_tied` to the number corresponding to the correct answer.

1. Reject the null / Data is consistent with the alternative hypothesis
2. Fail to reject the null / Data is consistent with the null hypothesis
3. Unable to tell using our staff confidence interval

Hint: If you're confused, take a look at [this chapter (https://www.inferentialthinking.com/chapters/13/4/using-confidence-intervals.html)](https://www.inferentialthinking.com/chapters/13/4/using-confidence-intervals.html) of the textbook.

```
BEGIN QUESTION
name: q2_4
manual: false
```

```
In [26]: restaurants_tied = 2 # SOLUTION
```

```
In [27]: # TEST
         1 <= restaurants_tied <= 3
Out[27]: True
```

```
In [28]: # HIDDEN TEST
         restaurants_tied == 2
Out[28]: True
```

Question 5

What if, instead, we use a P-value cutoff of 1%? Do we reject the null, fail to reject the null, or are we unable to tell using our staff confidence interval?

Assign `cutoff_one_percent` to the number corresponding to the correct answer.

1. Reject the null / Data is consistent with the alternative hypothesis
2. Fail to reject the null / Data is consistent with the null hypothesis
3. Unable to tell using our staff confidence interval

```
BEGIN QUESTION
name: q2_5
manual: false
```

```
In [29]: cutoff_one_percent = 2 # SOLUTION
```

```
In [30]: # TEST
         1 <= cutoff_one_percent <= 3
Out[30]: True
```

```
In [31]: # HIDDEN TEST
         cutoff_one_percent == 2
Out[31]: True
```

Question 6

What if we use a P-value cutoff of 10%? Do we reject, fail to reject, or are we unable to tell using our confidence interval?

Assign `cutoff_ten_percent` to the number corresponding to the correct answer.

1. Reject the null / Data is consistent with the alternative hypothesis
2. Fail to reject the null / Data is consistent with the null hypothesis
3. Unable to tell using our staff confidence interval

```
BEGIN QUESTION
name: q2_6
manual: false
```

```
In [32]: cutoff_ten_percent = 3 # SOLUTION
```

```
In [33]: # TEST
         1 <= cutoff_ten_percent <= 3

Out[33]: True
```

```
In [34]: # HIDDEN TEST
         cutoff_ten_percent == 3

Out[34]: True
```

3. Mid-Semester Survey

Once you have submitted, please also take the time to complete this class survey! The survey asks about how you're interacting with class resources now that everything online: we'll be using the information to adjust lectures, labs, office hours, and more for the rest of the semester.

The class survey is here: https://docs.google.com/forms/d/e/1FAIpQLSeif77zXdNmkEYTpFklQUZC6TxOd5Rd-mMmTQJDKqqwpxngSg/viewform?usp=sf_link (https://docs.google.com/forms/d/e/1FAIpQLSeif77zXdNmkEYTpFklQUZC6TxOd5Rd-mMmTQJDKqqwpxngSg/viewform?usp=sf_link)

Question 1. Fill out the class survey linked above. Right before submitting, a special string will be displayed. Set `special_string` to the special string at the end of the form.

```
BEGIN QUESTION
name: q3_1
manual: false
```

```
In [35]: special_string = "april fools" # SOLUTION
```

```
In [36]: # TEST
         special_string

Out[36]: 'april fools'
```

4. Submission

Once you're finished, select "Save and Checkpoint" in the File menu and then execute the `submit` cell below. The result will contain a link that you can use to check that your assignment has been submitted successfully. If you submit more than once before the deadline, we will only grade your final submission. If you mistakenly submit the wrong one, you can head to okpy.org (https://okpy.org/) and flag the correct version. To do so, go to the website, click on this assignment, and find the version you would like to have graded. There should be an option to flag that submission for grading!

```
In [37]: _ = ok.submit()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call l
ast)
<ipython-input-37-cc46ca874451> in <module>
----> 1 _ = ok.submit()

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

```
In [38]: # For your convenience, you can run this cell to run all the tests at on
         ce!
         import os
         print("Running all tests...")
         _ = [ok.grade(q[:-3]) for q in os.listdir("tests") if q.startswith('q')
         and len(q) <= 10]
         print("Finished running all tests.")
```

```
Running all tests...
Finished running all tests.
```