

# HW 08 Autograder

● Graded

Student

Sangwon Ji

Total Points

69 / 64 pts

Autograder Score

64.0 / 64.0

Passed Tests

Public Tests

Question 2

Early Submission Bonus

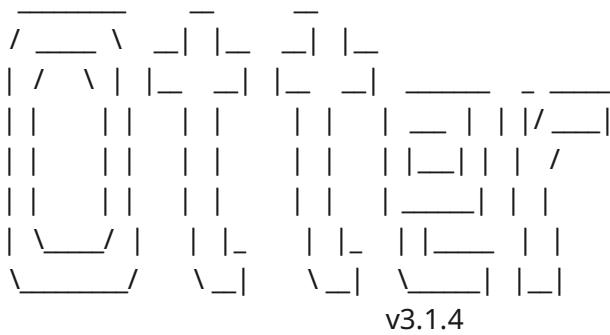
5 / 0 pts

✓ + 5 pts Early Submission Bonus

+ 0 pts No bonus

Autograder Results

Autograder Output



lower: 0.47506253911140456 upper: 0.5749374608885954  
With Michelle's sample size, you would predict a sample mean SD of 0.005000.  
With this smaller sample size, you would predict a sample mean SD of 0.049937  
With this larger sample size, you would predict a sample mean SD of 0.001579

----- GRADING SUMMARY -----

Error encountered while trying to verify scores with log:  
'TestCaseResult' object has no attribute 'hidden'

Successfully uploaded submissions for: sangwon@berkeley.edu

Total Score: 64.000 / 64.000 (100.000%)

	name	score	max_score
0	Public Tests	NaN	NaN
1	q2_1	6.0	6.0
2	q2_3	6.0	6.0
3	q2_5	6.0	6.0
4	q3_1	6.0	6.0
5	q3_3	6.0	6.0
6	q3_4	6.0	6.0
7	q3_5	6.0	6.0
8	q3_6	6.0	6.0
9	q3_7	5.0	5.0
10	q3_8	5.0	5.0
11	q3_9	4.0	4.0
12	q4_1	2.0	2.0

## Public Tests

q2\_1 results: All test cases passed!

q2\_3 results: All test cases passed!

q2\_5 results: All test cases passed!

q3\_1 results: All test cases passed!

q3\_3 results: All test cases passed!

q3\_4 results: All test cases passed!

q3\_5 results: All test cases passed!

q3\_6 results: All test cases passed!

q3\_7 results: All test cases passed!

q3\_8 results: All test cases passed!

q3\_9 results: All test cases passed!

q4\_1 results: All test cases passed!

## Submitted Files

In [1]:

```
# Initialize Otter
import otter
grader = otter.Notebook("hw08.ipynb")
```

## Homework 8: Sample Sizes and Confidence Intervals

### Helpful Resource:

- [Python Reference](#): Cheat sheet of helpful array & table methods used in Data 8!

### Recommended Readings:

- [Estimation](#)
- [Why the Mean Matters](#)

Please complete this notebook by filling in the cells provided. Before you begin, execute the following cell to setup the notebook by importing some helpful libraries. Each time you start your server, you will need to execute this cell again.

For all problems that you must write 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. Otherwise, you will fail tests that you thought you were passing previously!

### Deadline:

This assignment is due **Tuesday, 7/26 at 11:59pm PT**. Turn it in by Monday, 7/25 at 11:59pm PT for 5 extra credit points. Late work will not be accepted as per the [policies](#) page.

**Note: This homework has hidden tests on it. That means even though tests may say 100% passed, it doesn't mean your final grade will be 100%. We will be running more tests for correctness once everyone turns in the homework.**

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.

You should start early so that you have time to get help if you're stuck. Office hours are held Tuesday through Friday. The schedule appears on <http://data8.org/su22/office-hours.html>.

In [2]:

```
# 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)
```

## 1. Bounding the Tail of a Distribution

A community has an average age of 45 years with a standard deviation of 5 years.

In each part below, fill in the blank with a percent that makes the statement true **without further assumptions**, and explain your answer.

*Note:* No credit will be given for loose bounds such as "at least 0%" or "at most 100%". Give the best answer that is possible with the information given.

**Question 1.1.** At least \_\_\_% of the people are between 25 and 65 years old. (6 Points)

At least 93.75% of the people are between 25 and 65 years old. Using the Chebychev's Bounds, since we don't know the distribution, not an exact answer, but we can get a bound. In a standard deviation of 5 years, 65-45, 20, dividing that by 5, we have average  $\pm 4$  Sds. In a formula  $1 - (1/z^{**2})$ , 4 for z, we get 0.9375, 93.75 percent.

**Question 1.2.** At most \_\_\_% of the people have ages that are not in the

range 25 years to 65 years. **(6 Points)**

At most 6.25% of the people have ages that are not in the range 25 years to 65 years. This is because we know that the bound of percentage of people in range of 25 years to 65 years old is 93.75. So, from the total, 100 percent, we subtract the number, and we will be able to get the population outside of the range. That is  $100 - 93.75 = 6.25$ .

**Question 1.3.** At most \_\_\_% of the people are more than 65 years old. **(6 Points)**

*Hint:* If you're stuck, try thinking about what the distribution may look like in this case.

At most 6.25% of the people are more than 65 years old. we know that the range that is not in the 25 to 65 range are 6.25%. In this case, talking about there is no value located under 25, there is a chance that it is at most 6.25%, which is all the population outside of 25 to 65 range.

## 2. Sample Size and Confidence Level

A data science class at the large Data 8 University wants to estimate the percent of Facebook users among students at the school. To do this, they need to take a random sample of students. You can assume that their method of sampling is equivalent to drawing at random with replacement from students at the school.

**Please review [Section 14.6](#) of the textbook before proceeding with this section. You will be able to understand and solve the problems more efficiently!**

**Question 2.1.** Assign  to the smallest number of students they should sample to ensure that a **95%** confidence interval for the parameter has a width of no more than 6% from left end to right end. **(6 points)**

*Hint:* How can our data be represented to show if a student in the sample is a Facebook user? Given this, what assumptions can we make for the SD of the population? [Section 14.6](#) might be helpful!

*Note:* While the true smallest sample size would have to be an integer, please leave your answer in decimal format for the sake of our tests.

In [3]: `smallest = (4 * (0.5 / 0.06)) ** 2`  
`smallest`

Out [3]: 1111.1111111111113

In [4]: `grader.check("q2_1")`

Out [4]: q2\_1 results: All test cases passed!

**Question 2.2.** Suppose the data science class decides to construct a 90% confidence interval instead of a 95% confidence interval, but they still require that the width of the interval is no more than 6% from left end to right end. Will they need the same sample size as in 2.1? Pick the right answer and explain further without calculation. **(6 Points)**

1. Yes, they must use the same sample size.
2. No, a smaller sample size will work.
3. No, they will need a bigger sample.

2: No, a smaller sample size will work. This is because in 95% confidence interval, we calculate the width with 4SDs, and using 90 percent confidence interval would have a smaller sample size since we are using less SDs in a calculation. In this essence, the sample size will go down.

**Question 2.3.** The professor tells the class that a 90% confidence interval for the parameter is constructed exactly like a 95% confidence interval, except that you have to go only 1.65 SDs on either side of the estimate ( $\pm 1.65$ ) instead of 2 SDs on either side ( $\pm 2$ ). Assign `smallest_num` to the smallest number of students they should sample to ensure that a **90%** confidence interval for the parameter has a width of no more than 6% from left end to right end. **(6 points)**

*Note:* While the true smallest sample size would have to be an integer, please leave your answer in decimal format for the sake of our tests.

In [5]: `smallest_num = (1.65 * 2 * (0.5 / 0.06)) ** 2`  
`smallest_num`

Out [5]: 756.25

In [6]: `grader.check("q2_3")`

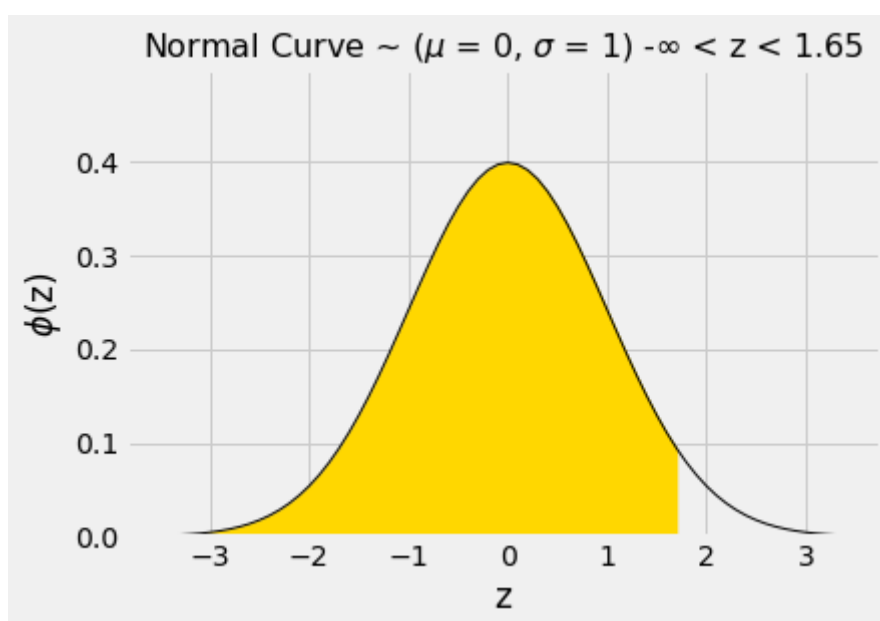
Out [6]: q2\_3 results: All test cases passed!

For this next exercise, please consult [Section 14.3.4](#) of the textbook for similar examples.

The students are curious about how the professor came up with the value 1.65 in Question 2.3. She says she ran the following two code cells. The first one calls the `datascience` library function `plot_normal_cdf`, which displays the proportion that is at most the specified number of SDs above average under the normal curve plotted with standard units on the horizontal axis. You can find the documentation [here](#).

*Note:* The acronym `cdf` stands for `cumulative distribution function`. It measures the proportion to the left of a specified point under a probability histogram.

In [7]: `plot_normal_cdf(1.65)`



To run the second cell, the professor had to first import a Python library for probability and statistics:

In [8]: 

```
# Just run this cell
from scipy import stats
```

Then she used the `norm.cdf` method in the library to find the gold proportion above.



In [9]:

```
# Just run this cell
stats.norm.cdf(1.65)
```

Out [9]: 0.9505285319663519

*Note:* You do not need to understand how the `scipy` library or how to use the method yourself.

**Question 2.4.** This shows that the percentage in a normal distribution that is at most 1.65 SDs above average is about **95%**. Explain why 1.65 is the right number of SDs to use when constructing a **90%** confidence interval. **(6 Points)**

The percentage of 1.65 SD, you can get by subtracting, 100-95 which is 5 percent. The percent below 1.65SD will also be 5 percent. The percentage in the middle is 90 percent, between above 1.65 SD and below 1.65 SD. That will sum up to a 100%, which indicates that 1.65 is the correct when using for constructing 90% confidence interval.

In [10]:

```
# Just run this cell, do not change it.
stats.norm.cdf(2.33)
```

Out [10]: 0.9900969244408357

**Question 2.5.** The cell above shows that the proportion that is at most 2.33 SDs above average in a normal distribution is 99%. Assign `option` to the right option to fill in the blank: **(6 points)**

If you start at the estimate and go 2.33 SDs on either side, then you will get a \_\_\_% confidence interval for the parameter.

1. 99.5
2. 99
3. 98.5
4. 98

In [11]:

```
option = 4
option
```

Out [11]: 4

```
In [12]: grader.check("q2_5")
```

Out [12]: q2\_5 results: All test cases passed!

### 3. Polling and the Normal Distribution

```
In [13]: # 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)
```

Michelle is a statistical consultant, and she works for a group that supports Proposition 68 (which would mandate labeling of all horizontal and vertical axes) called Yes on 68. They want to know how many Californians will vote for the proposition.

Michelle polls a uniform random sample of all California voters, and she finds that 210 of the 400 sampled voters will vote in favor of the proposition. We have provided a table for you below which has 3 columns: the first two columns are identical to `sample`. The third column contains the proportion of total voters that chose each option.

```
In [14]: sample = Table().with_columns(
    "Vote", make_array("Yes", "No"),
    "Count", make_array(210, 190))

sample_size = sum(sample.column("Count"))
sample_with_proportions = sample.with_column("Proportion",
    sample.column("Count") / sample_size)
sample_with_proportions
```

Out [14]:

Vote	Count	Proportion
Yes	210	0.525

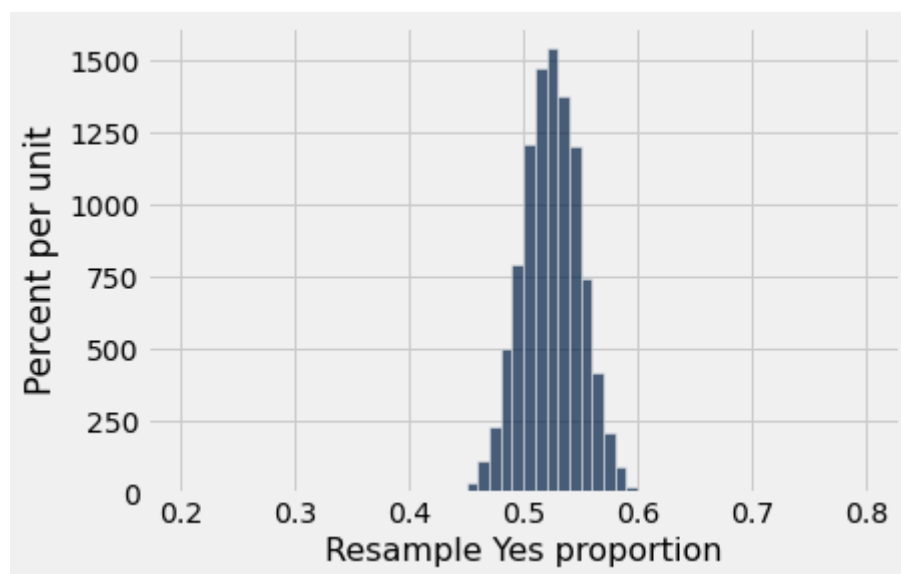
**Question 3.1.** Michelle wants to use 10,000 bootstrap resamples to compute a confidence interval for the proportion of all California voters who will vote Yes.

Fill in the next cell to simulate an empirical distribution of Yes proportions. Use bootstrap resampling to simulate 10,000 election outcomes, and assign `resample_yes_proportions` to contain the Yes proportion of each bootstrap resample. Then, visualize `resample_yes_proportions` with a histogram. **You should see a bell shaped curve centered near the proportion of Yes in the original sample. (6 points)**

*Hint:* `sample_proportions` may be useful here!

In [15]:

```
resample_yes_proportions = make_array()
for i in np.arange(10000):
    resample = sample_proportions(400, make_array(210/400, 190/400)).item(0)
    resample_yes_proportions = np.append(resample_yes_proportions, resample)
Table().with_column("Resample Yes proportion",
resample_yes_proportions).hist(bins=np.arange(.2, .8, .01))
```



In [16]:

```
grader.check("q3_1")
```

Out [16]: q3\_1 results: All test cases passed!

**Question 3.2.** Why does the Central Limit Theorem (CLT) apply in this situation, and how does it explain the distribution we see above? **(6 points)**

According to the Central Limit Theorem (CLT), as the sample size simulation was done 10000 times, a large amount of number, the distribution of random variable  $x$  is approaching a normal distribution. So, we can first not think about the distribution of the population where this sample is drawn. The distribution shown above, the proportion of people who voted yes is close to normal and shows a bell shaped graph. This case the CLT is applied which explains how the distribution is close to normal.

In a population whose members are 0 and 1, there is a simple formula for the **standard deviation of that population**:

$$\text{standard deviation of population} = \sqrt{(\text{proportion of 0s}) \times (\text{proportion of 1s})}$$

(Figuring out this formula, starting from the definition of the standard deviation, is a fun exercise for those who enjoy algebra.)

**Question 3.3.** Using only the Central Limit Theorem and the numbers of Yes and No voters in our sample of 400, *algebraically* compute the predicted standard deviation of the `resample_yes_proportions` array. Assign this number to `approximate_sd`. **Do not access the data in `resample_yes_proportions` in any way. (6 points)**

Remember that the standard deviation of the sample means can be computed from the population SD and the size of the sample (the formula above might be helpful). If we do not know the population SD, we can use the sample SD as a reasonable approximation in its place. [This section](#) of the textbook also may be helpful.

```
In [17]: approx_pop_sd = ((0.525 * 0.475)) ** 0.5
         approximate_sd = approx_pop_sd/(400 ** 0.5)
         approximate_sd
```

```
Out [17]: 0.024968730444297725
```

```
In [18]: grader.check("q3_3")
```

```
Out [18]: q3_3 results: All test cases passed!
```

**Question 3.4.** Compute the standard deviation of the array `resample_yes_proportions`, which will act as an approximation to the true SD of

the possible sample proportions. This will help verify whether your answer to question 3.3 is approximately correct. **(6 points)**

```
In [19]: exact_sd = np.std(resample_yes_proportions)
exact_sd
```

```
Out [19]: 0.025089200221410014
```

```
In [20]: grader.check("q3_4")
```

```
Out [20]: q3_4 results: All test cases passed!
```

**Question 3.5. Again, without accessing `resample_yes_proportions` in any way,** compute an approximate 95% confidence interval for the proportion of Yes voters in California. **(6 points)**

The cell below draws your interval as a red bar below the histogram of `resample_yes_proportions`; use that to verify that your answer looks right.

*Hint:* How many SDs corresponds to 95% of the distribution promised by the CLT? Recall the discussion in the textbook [here](#).

*Hint:* The `approximate_sd` variable you previously defined may be helpful!

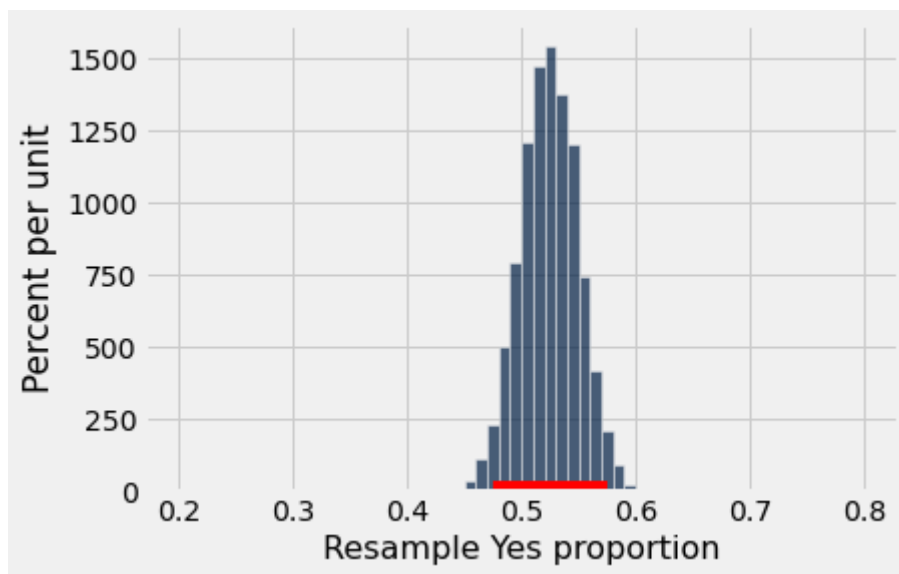
```
In [21]: lower_limit = 0.525 - 2 * (approximate_sd)
upper_limit = 0.525 + 2 * (approximate_sd)
print('lower:', lower_limit, 'upper:', upper_limit)
```

```
lower: 0.47506253911140456 upper: 0.574937460885954
```

```
In [22]: grader.check("q3_5")
```

```
Out [22]: q3_5 results: All test cases passed!
```

```
In [23]: # Run this cell to plot your confidence interval.
Table().with_column("Resample Yes proportion",
resample_yes_proportions).hist(bins=np.arange(.2, .8, .01))
plt.plot(make_array(lower_limit, upper_limit), make_array(0, 0), c='r', lw=10);
```



Your confidence interval should overlap the number 0.5. That means we can't be very sure whether Proposition 68 is winning, even though the sample Yes proportion is a bit above 0.5.

The Yes on 68 campaign really needs to know whether they're winning. It's impossible to be absolutely sure without polling the whole population, but they'd be okay if the standard deviation of the sample mean were only 0.005. They ask Michelle to run a new poll with a sample size that's large enough to achieve that. (Polling is expensive, so the sample also shouldn't be bigger than necessary.)

Michelle consults Chapter 14 of your textbook. Instead of making the conservative assumption that the population standard deviation is 0.5 (coding Yes voters as 1 and No voters as 0), she decides to assume that it's equal to the standard deviation of the sample,

$$\sqrt{(\text{Yes proportion in the sample}) \times (\text{No proportion in the sample})}.$$

Under that assumption, Michelle decides that a sample of 9,975 would suffice.

Does Michelle's sample size achieve the desired standard deviation of sample means? What SD would you achieve with a smaller sample size? A higher sample size?

**Question 3.6.** To explore this, first compute the SD of sample means obtained by using Michelle's sample size. **(6 points)**

In [24]:

```
estimated_population_sd = ((0.525) * (0.475)) ** 0.5
```

```
michelle_sample_size = 9975
michelle_sample_mean_sd = estimated_population_sd/ (michelle_sample_size **
0.5)
print("With Michelle's sample size, you would predict a sample mean SD of %f."
% michelle_sample_mean_sd)
```

With Michelle's sample size, you would predict a sample mean SD of 0.005000.

In [25]: `grader.check("q3_6")`

Out [25]: q3\_6 results: All test cases passed!

**Question 3.7.** Next, compute the SD of sample means that you would get from a smaller sample size. Ideally, you should pick a number that is significantly smaller, but any sample size smaller than Michelle's will do. **(5 points)**

```
smaller_sample_size = 100
smaller_sample_mean_sd = estimated_population_sd/ (smaller_sample_size **
0.5)
print("With this smaller sample size, you would predict a sample mean SD of
%f" % smaller_sample_mean_sd)
```

With this smaller sample size, you would predict a sample mean SD of 0.049937

In [27]: `grader.check("q3_7")`

Out [27]: q3\_7 results: All test cases passed!

**Question 3.8.** Finally, compute the SD of sample means that you would get from a larger sample size. Here, a number that is significantly larger would make any difference more obvious, but any sample size larger than Michelle's will do. **(5 points)**

```
larger_sample_size = 100000
larger_sample_mean_sd = estimated_population_sd/ (larger_sample_size ** 0.5)
print("With this larger sample size, you would predict a sample mean SD of %f"
% larger_sample_mean_sd)
```

With this larger sample size, you would predict a sample mean SD of 0.001579

In [29]: `grader.check("q3_8")`

Out [29]: q3\_8 results: All test cases passed!

**Question 3.9.** Based off of this, was Michelle's sample size approximately the minimum sufficient sample, given her assumption that the sample SD is the same as the population SD? Assign `min_sufficient` to `True` if 9,975 was indeed approximately the minimum sufficient sample, and `False` if it wasn't. (4 points)

In [30]: `min_sufficient = True`  
`min_sufficient`

Out [30]: True

In [31]: `grader.check("q3_9")`

Out [31]: q3\_9 results: All test cases passed!

## 4. Mid-Semester Survey

Once you have submitted, please also take the time to complete the Mid-Semester Survey! We really appreciate your honest feedback and it helps us improve the course!

The Mid-Semester survey is linked here:

<https://forms.gle/HWxNuB2fs4gmw3AK7>

**Question 4.1.** Fill out the mid-semester survey linked above. Once you have submitted, a secret word will be displayed. Set `secret_word` to the secret string at the end of the form. (2 points)

In [35]: `secret_word = "doggo8"`

In [36]: `grader.check("q4_1")`

Out [36]: q4\_1 results: All test cases passed!

You're done with Homework 8!



**Important submission steps:** 1. Run the tests and verify that they all pass. 2. Choose **Save Notebook** from the **File** menu, then **run the final cell**. 3. Click the link to download the zip file. 4. Go to [Gradescope](#) and submit the zip file to the corresponding assignment. The name of this assignment is "HW 08 Autograder".

**It is your responsibility to make sure your work is saved before running the last cell.**

---

To double-check your work, the cell below will rerun all of the autograder tests.

In [37]:

```
grader.check_all()
```

Out [37]:

q2\_1 results: All test cases passed!

q2\_3 results: All test cases passed!

q2\_5 results: All test cases passed!

q3\_1 results: All test cases passed!

q3\_3 results: All test cases passed!

q3\_4 results: All test cases passed!

q3\_5 results: All test cases passed!

q3\_6 results: All test cases passed!

q3\_7 results: All test cases passed!

q3\_8 results: All test cases passed!

q3\_9 results: All test cases passed!

q4\_1 results: All test cases passed!

## Submission

Make sure you have run all cells in your notebook in order before running the cell below, so that all images/graphs appear in the output. The cell below will generate a zip file for you to submit. **Please save before exporting!**


---

In [38]:

```
# Save your notebook first, then run this cell to export your submission.  
grader.export(pdf=False)
```


<IPython.core.display.HTML object>

▼ .OTTER\_LOG

 Download

1	Binary file hidden. You can download it using the button above.
---	---

▼ \_\_zip\_filename\_\_

 Download

1	<a href="#">hw08_2022_07_25T15_21_07_110871.zip</a>
---	---