Problem Set 4: Sea Level Rise

The questions below are due on Friday May 06, 2022; 11:36:00 AM.
Checkoff start: Apr 29 at 09:00AM
Checkoff due: May 06 at 09:00PM

Download Files

Pset Buddy

Your buddy for this week is kateell

Go ahead and reach out to your buddy via their kerb/email. Additionally, watch out for an email from your buddy! Please check this box once you have successfully made contact with each other. If your buddy has not responded after 2 days, please start the pset and contact the course staff. This will not affect grading.

- [x] I have made contact with my buddy

View Answer **100.00%**

You have infinitely many submissions remaining.

Introduction

In this pset, you will code a Monte Carlo simulation of several decades of future flood damage due to sea level rise and compare alternative strategies involving some combination of 'prepare', 'repair', and 'retreat' strategies. While the problem statement and risk management strategies are simplified, they provide the ground framework for addressing some of the very real issues associated with determining who should bear the cost of property damage along our coastlines and what regulation of coastal land use is legal, appropriate, and equitable. If any of this truly speaks to you or if you're just curious to learn more, please check out the linked academic papers and newspaper articles referencing related 'real world' settings at the end of the pset.

Background

In the U.S., approximately 40% of the population lives in coastal urban areas. Many coastal cities face the risk of rising sea levels (see *Links about sea level rise estimates and visualizations* in the References section). Flooding, in particular, causes an estimated average cost of $4.4 billion per event. Boston ranks 8th worldwide based on estimated economic loss due to coastal flooding. The National Oceanic and Atmospheric Administration (NOAA) estimates that Boston urban and coastline areas face a near 100% risk of at least one flood over 6 ft from now (2021) to 2050. Flooding risk assessments serve various purposes involving regional planning, management of critical infrastructure vulnerability (transportation, energy, telecommunication), policymaking (housing, population, social/economic development), and emergency response.

One common approach in flood management is to use Monte Carlo Simulation (MCS) to estimate the flooding risk as probabilistic distributions. Often, we estimate an annual average sea-level rise and its 95% confidence interval to approximate plausible lower and upper bounds during future years under various policy and decision-making scenarios. In addition, flooding risk management and adaptation actions involve multiple stakeholders with conflicts and competing interests. Therefore, scenario-based simulation often assists the decision-making and negotiation process, so decision-makers can compare the outcomes, discuss the trade-offs, and identify a better solution. In this way, simulation tools can estimate long-term consequences based on different decision scenarios while taking some of the uncertainty into account.

It is worth clarifying that we provide an extremely simplified version for the sake of this pset. In reality, flooding risk modeling is much more complex with additional factors, such as multiple storm events, winds, tides, waves, extreme weather, land surface changes, and interdependence across years. While such complexities create additional uncertainty, the basic approach is illustrated in this exercise.

Problem Description

There are two subproblems that we would like to implement:

1. Simulating annual sea level risk out to 2100.
2. Simulating annual cost of damage due to sea level rise, and understanding the effect that certain prevention measures can have.

0. Installing Required Packages

Before you start, you may need to install a number of python packages, listed in `requirements.txt`. These can be installed by running the following command in the console: `pip install -r requirements.txt`

1. Estimating Sea Level Rise (SLR)

In sub-problem 1, you will first write a function for generating a plot of probabilistic flood distributions based on the annual average sea level (SL). Table 1 summarizes the sea level rise (SLR) forecasts along coastal areas from 2020 to 2100 for slow SLR and fast SLR scenarios. In this pset, we consider the range of heights for the slow- and fast-rise scenarios to be our approximation for the 95% confidence interval of likely sea-level rise. Note that we use a 95% confidence interval because any observed data following a normal distribution will have 95% of its samples falling within 2 standard deviations of the mean. Please Google this if this is not intuitive to you!

Table 1. Sea level rise projection based on slow SLR and fast SLR scenarios.

Year	Water Level - Slow SLR (ft)	Water Level - Fast SLR (ft)
2020	3.9	4.4
2030	4.1	4.8
2040	4.3	5.3
2050	4.5	5.9
2060	4.7	6.6
2070	4.9	7.3
2080	5.0	8.1
2090	5.2	9.0
2100	5.4	10.0

Hints

- The following numpy quickstart guide may be helpful for part 1 of this pset https://numpy.org/devdocs/user/quickstart.html, particularly the "Vector Stacking" and "Universal Functions" sections.

Function 1.1a: Interpolate the Projected Annual Sea Level Rise

Assume the actual annual sea level rise is normally distributed with no dependencies from one year to the next, and with 2.5 and 97.5 percentiles as indicated by the ‘low’ and ‘high’ estimates in Table 1 for the start of each decade out to 2100. Interpolate the mean (average of the 2.5 percentile and 97.5 percentile values) and standard deviation of each year’s normal probability distribution for the sea level rise for the 81 years from 2020 to 2100.

Implement `predicted_sea_level_rise()`: Calculate (or lookup, if it lands on the decade) the interpolated values across 81 years from 2020 to 2100 for the mean of predicted sea level raises. Optionally, if `show_plot = True`, plot the results with a 95% confidence interval. Return a numpy array with each row representing a year's data where there are 5 columns: year, mean, lower, upper, and std. As a reminder, lower represents the lower bound 2.5 percentile value for sea level raise, and upper represents the 97.5 percentile upper bound.

A plot of the results should look like Figure 2.1a below.

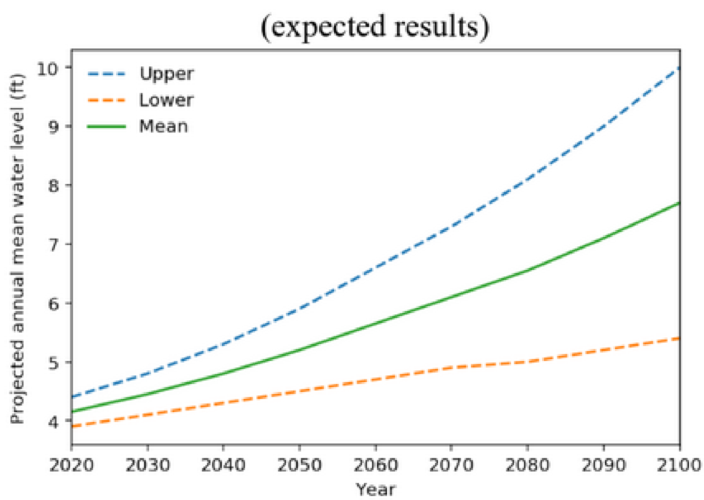

Figure 2.1a Time-series of projected annual average water level and 95% confidence interval.

Hints

- If you want just a column of a numpy array, one convenient way is: `x[:, 0]`, try it! Intuitively, this indexing notation tells us to take elements from the array `x` where everything along the 1st dimension (rows) is included, and only the 0th element along the 2nd dimension (columns) is included.

Function 1.2a: Simulate a Specific Year’s Sea Level Rise

Assume the actual sea level rise during each year is normally distributed (independent of what happens in all other years) with the means and standard deviations that you estimated in `predicted_sea_level_rise()`. Fill in `simulate_year()` to simulate a particular instance of sea-level rise during any particular year. This function takes the output from `predicted_sea_level_rise()` and uses Monte Carlo simulation to draw a random sample based on the assumed normal probability distribution.

Hints

- Check out numpy's `random.normal`. Read through the docs and make sure you understand what the inputs and outputs should look like.
- Numpy aside: If you're feeling adventurous and would like to write code that is more in the spirit of using all of Numpy's fancy optimizations (extra motivation: these functions and broadcasting operations are commonly used in ML/AI contexts for their speed), check out the documentation and examples on the `numpy.where` function. An alternative is just to go through each of the rows one at a time to identify the correct year's information.

Function 1.2b: Plot Monte-Carlo Sea Level Rise Data

Implement `plot_mc_simulation()`: Use `simulate_year()` to run 500 simulations of the 81 years out to 2100 and visualize your results in a plot that should look like Figure 2.1b.

Hints:

- Take a look at matplotlib's pyplot. In particular, `plot()` will give you a line connecting data points, and `scatter()` will plot individual points that you specify. For ensuring that your plot is readable, make sure to check out the documentation for axis labels, legends, and plot titles. Use `show()` to visualize the plot you created. Play around with parameters for the line and scatter plots, like font size and line styles to see what looks best! Note that your graphs don't have to look exactly like the ones we have as reference.
- Make sure you understand exactly what inputs pyplot's `plot` and `scatter` methods expect, and what their shapes should be. When in doubt, look up basic toy examples online.

(expected results)

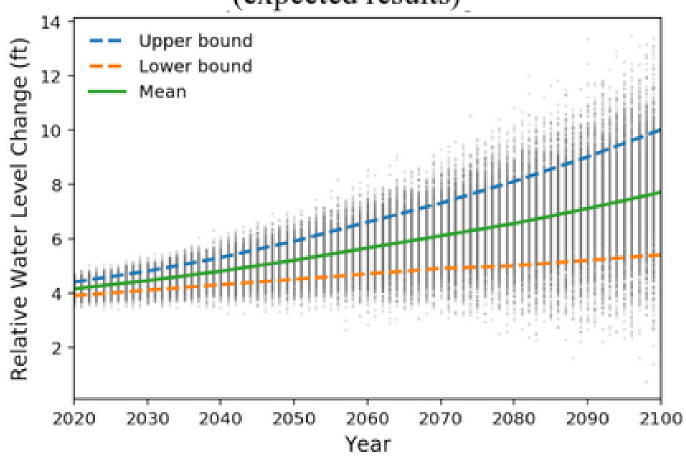

Figure 2.1b Simulated projected sea level change in feet from 2020 to 2100.

2. Estimating damage resulting from 1-10 feet sea level rise

In this subproblem, we focus on a hypothetical individual homeowner considering alternative 'repair, prepare, or retreat' strategies to protect their coastal houses from sea level rise. Suppose a house has a current market value of **V = $400,000 for the building**. For simplicity, we'll assume that all values are expressed in inflation-adjusted dollars and there is no expectation of property appreciation (or decline) out to 2100 (think of these numbers as inflation-adjusted values). We assume this homeowner faces zero damage risk if the sea level rise is less than 5 feet and increasing damage cost if the sea level rise is more than 5 feet. The amount of damage is shown in Table 2.2a (as a percentage of the building value, remember to divide by 100!) and is the damage that is expected for houses that have *not invested in any flood damage prevention measures*.

Table 2.2a Estimated damage loss (in percent) for each sea-level rise (when no prevention measures have been made).

Water Level (ft)	Property Damage Loss (% of building value)
5	0
6	10
7	25
8	45
9	75
10	100

Function 2.1: Simulated Water Level Estimate

Implement `water_level_est()`: Simulate the water level for each year according to the data we originally gathered in `predicted_sea_level_rise()`. Note that we expect just a list of floats representing each year's sampled water level from 2020 to 2100. You may want to reuse `simulate_year()`.

Function 2.2a: Damage Cost Estimate

Implement `repair_only()`: Calculate the cost per year in potential property damage given:

- Simulated water levels per year from `water_level_est()`
- The estimated property damage percentages as seen in Table 2.2b (stored inside the parameter `water_level_loss_no_prevention`)
- House value V = $400,000

If you wish to visually debug, plot your results! For reference, your plot should look like Figure 2.2a.

Hints:

- Import and use the `scipy.interpolate.interp1d` class to interpolate the {water level → property damage percentage} function using values from `water_level_loss_no_prevention`. Use `fill_value="extrapolate"`. What object type does this class return and how would you use it? (Additional documentation for this class can be found here https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.interp1d.html).
- Remember to return each year's cost in thousands.
- We assume that the cost never exceeds the house's value.
- You shouldn't perform Monte Carlo simulation here, all you need is interpolation. Apologies if the sanity check figure below is a bit misleading.

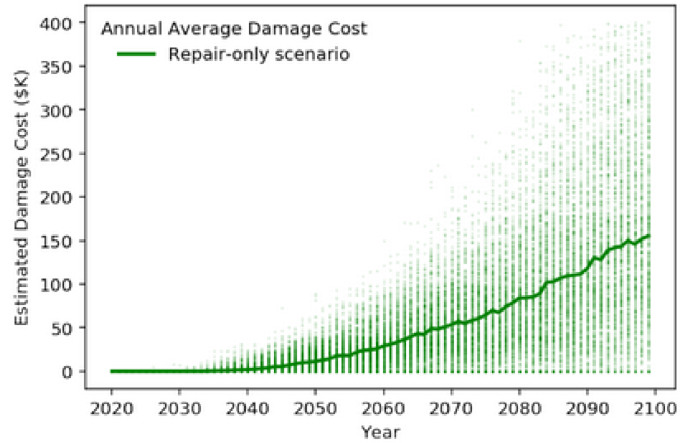

Figure 2.2a Simulated annual damage cost for our house, 2020 – 2100.

Function 2.2b. Damage Cost Estimate Based on a Simple Prevention Strategy

You'll find that `water_level_est()` indicates a high likelihood of substantial, recurring losses from SLR. Thus, the homeowner is likely to consider (or their insurer is likely to require!) some investment in flood protection measures. Suppose a one-time investment of Z=$200000 could reduce the damage estimates in Table 2.2a so the loss percentages would match the levels shown in Table 2.2b.

Water Level (ft)	Property Damage Loss (% of building value) IF damage prevention measure is in place
5	0
6	5
7	15
8	30
9	70
10	100

Table 2.2b Estimated damage cost (in percentage, with prevention) by sea-level rise.

Let's evaluate a strategy whereby the homeowner invests no money in prevention until the first year in which SLR damage is at least T=$100000. In this case the homeowner pays for the $100000+ in repairs and invests an additional Z dollars to reduce future damage from SLR. **Note: the Z dollars invested do NOT count towards the damage costs! This is just something a homeowner should carefully consider in their prevention efforts! (You do not need to use Z in the function.)**

Implement `wait_a_bit()`: Similar to `repair_only()`, except now the property damage percentage depends on whether active prevention is being practiced. Calculate the costs incurred using this strategy against the water level rises that are provided.

Function 2.2c. Damage Cost Estimate Based on Immediate Prevention

Now, we envision a strategy where the homeowner takes SLR extremely seriously and immediately begins preparations for any potential property damage risks. Implement `prepare_immediately()` to calculate projected costs. This should be very similar to `repair_only()`. Again, no investment costs need to be considered in the damage costs. Assume that the investment considerations are tangential to our cost estimates.

Function 2.3a. Plot Prevention Strategies

Now we wish to compare the ‘wait a bit’ strategy and the two simpler strategies of ‘repair only’ or ‘prepare immediately’. Modify your MCS to simulate the 81-year distribution of annual costs under the three strategies. Which of the three strategies is expected to yield the lowest total cost over the entire time horizon? What is the average total cost of your 500 simulations for each of the three strategies (including the cost of the prevention measures as well as the damage losses)?

Implement `plot_strategies()`: Use 500 simulations of the ‘repair only’, ‘wait a bit’, and ‘prepare immediately’ strategies to generate a scatterplot of the annual damage losses for each year from 2020 to 2100 and overlay a line plot of the annual mean damage losses. This plot should look something like the plot in Figure 2.2b. If you've made it this far, congrats! You have successfully analyzed in a simplified setting the impact of certain prevention strategies on simulated SLR. Hopefully after this pset, you can appreciate how this can be extrapolated to real-life SLR scenarios - where much more extensive analysis is done to arrive at "optimal" solutions that must carefully weigh potential costs and benefits.

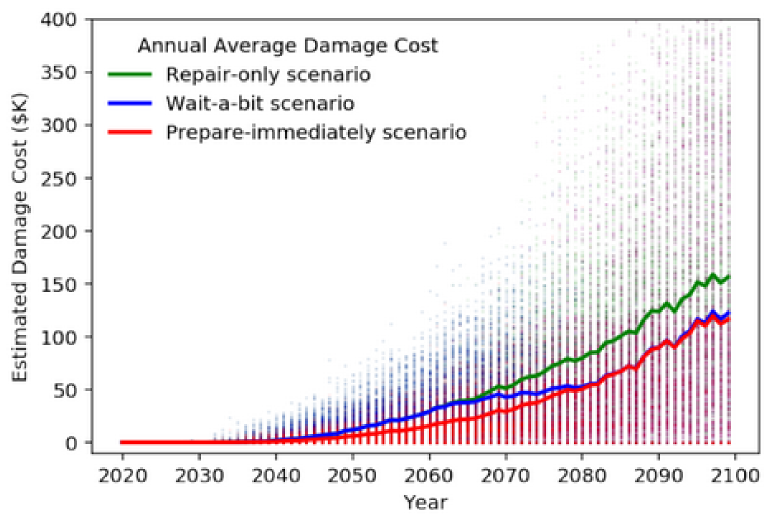

Figure 2.2b Property damage cost comparison

3. Simulating Sea Level Rise in the Cambridge Area

In this section, we will generate an animated map of the Cambridge area showing the properties that would be underwater at different flood levels.

Uncomment the remaining content included in the `if __name__ == '__main__'` section at the end of the file. Running this section should generate a new file, `animation.gif`, which contains the animation for flooding in the Cambridge area.

Note: The gif may not render in your plot’s tab. However, `animation.gif`, should appear in the same folder containing `ps4.py`. If opening the file with your machine’s default settings does not create an animation, right click on the “animation.gif” file and “Open With” a browser that supports GIF (Chrome, Firefox, Safari,etc.)

Be sure to keep a copy of the animation generated, as well as other plots generated in earlier parts. You will be expected to present it during your checkoff.

Hand-in Procedure

Time and Collaboration Info

At the start of each file in a comment, write down the names of the people you collaborated with. For example:

```
# Problem Set 4
# Name: Jane Lee
# Collaborators: John Doe
```

Please estimate the number of hours you spent on the Problem Set in the question box below.

12

Thank you!

You have infinitely many submissions remaining.

Submission

Be sure to run the student tester and make sure all the tests pass. However, the student tester contains only a subset of the tests that will be run to determine the problem set grade. Passing all of the provided test cases does not guarantee full credit on the pset. Make sure that any calls you make to different functions are under `if __name__ == '__main__'`.

You may upload new versions of each file until Apr 28 at 09:00PM, but anything uploaded after that time will be counted towards your late days, if you have any remaining. If you have no remaining late days, you will receive no credit for a late submission.

When you upload a new file your old one will be overwritten.

Download Most Recent Submission

Select File No file selected

Code Submitted Successfully!

You have infinitely many submissions remaining.

References

Data Sources

Relative Sea Level Trend in Boston, Massachusetts, NOAA

Boston open data related to flooding

Links about sea level rise estimates and visualizations

New York Times article (April 24, 2016): What Could Disappear

WBUR News (March 17, 2018): Report: Boston Sea Level Projected To Rise 1.5 Feet By 2050

Aerts, Jeroen CJH, Ning Lin, Wouter Botzen, Kerry Emanuel, and Hans de Moel. "Low- probability flood risk modeling for New York City." Risk Analysis 33, no. 5 (2013): 772- 788. <<< This article demonstrates flooding risk modeling in detail, with digital elevation models (DEM) and inundation maps. >>>

Science News (August 6, 2019): With nowhere to hide from rising seas, Boston prepares for a wetter future

Climate Ready Boston Explore (interactive data viz)

Sea Change Boston by Sasaki (interactive data viz)

Boston underwater: How the rising sea levels will affect the city (interactive data viz)

Sea Level Rise organization: Massachusetts' Sea Level Is Rising, and It's Costing Over $1 Billion

Links about Property Insurance, Premiums, and Building Regulations:

New York Times article (2017): A Broke, and Broken, Flood Insurance Program <<< Key findings: Some critical problems in National Flood Insurance Program, including a data visualization comparing premiums collected vs. losses paid >>>

Dixon, Lloyd, Noreen Clancy, Bruce Bender, Aaron Kofner, David Manheim, and Laura Zakaras. Flood Insurance in New York City Following Hurricane Sandy. Rand Corporation, 2013. <<< Key findings: Approximately 3.6 percent of New York City structures are in the high-risk areas defined by the 2007 flood map, and most of the structures (72 percent) are homes for one to four families. Approximately 55 percent of one- to four-family homes in high-risk areas had federal flood insurance on the eve of Hurricane Sandy; approximately three-quarters of one- to four-family homes in the high-risk areas on the 2007 map are subject to the mandatory purchase requirement.>>>

Links about Equity and Social Policy:

NPR News: Business of Disaster: Insurance Firms Profited $400 Million After Sandy <<<Are coastal homeowners systematically underpaid by insurance companies after Sandy?>>>

ESRI, "Aftermath of Katrina: A Time of Environmental Racism," A story map about environmental justice showing that minority communities are more likely living at high risk and in underserved areas

Springer Nature, 90, 453-473 (2008): Kirshen, P., Knee, K., & Ruth, M. (2008). Climate change and coastal flooding in Metro Boston: Impacts and adaptation strategies. Climatic Change, 90(4), 453-473.

Shi, Linda, Boston Globe Op-Ed (March 5, 2020): "The Fiscal Challenges of Climate Change: Municipalities depend on the revenue of coastal development. That has to change."

Shi, Linda, and Andrew Vanuzzo, "Surging seas, rising fiscal stress: Exploring municipal fiscal vulnerability to climate change," Cities, Vol. 100, May 2020

MIT DUSP Climate Action Plan

MIT Course 11+6 (Urban Science and Planning with Computer Science)

This problem set was made in partnership with Course 11, Urban Studies and Planning. Do you like these sorts of problems that can involve both significant knowledge of computing and data analysis, and a deeper understanding of social choice, governance, urban planning, and regulation?

Visit MIT Course 11-6: Urban Science and Planning with Computer Science or email urban- science@mit.edu to speak to an advisor.