1 Bayesian Modeling

Bayes' rule tells us everything we need to know about updating our beliefs based on evidence. All strong forms of statistical inference are based on Bayes' rule.

To take an arbitrary model, fit it, and test it using Bayes' rule, we use Bayesian modeling. In Python, the most common tool for doing this is PyMC. This week you are going to use PyMC to model a system and fit it to observed data.

2 The Problem

There is a bay with one outlet to the ocean. At high tide, there is a lot of water in the bay. As the tide goes down, water rushes out of the bay. Jellyfish tend to drift in and out with the tides. (High tide occurs every 12 hours.)

In the outlet, GSU has placed a laser jellyfish counter. When you press the button on it, it counts all the jelly fish entering and exiting the bay for 15 minutes. At the end of 15 minutes it gives you a net count: how many more jellyfish are in the bay than there were when the count started.

The team at the bay has emailed you a CSV with some times that they started counts and the corresponding count:

```
timestamp,jellyfish_entering
2024-02-13T12:00,37
2024-02-13T13:02,38
2024-02-13T13:57,67
2024-02-13T14:55,106
2024-02-13T15:38,45
2024-02-13T17:16,56
2024-02-13T18:50,-33
2024-02-13T19:30,-55
2024-02-13T21:02,-104
2024-02-13T21:59,-64
2024-02-13T22:48,-48
2024-02-13T23:56,-7
```

They mention that the first measurement was made at the lowest possible tide.

They have asked you to come up with a formula for predicting how many jellyfish will be counted at any time. That is, your formula will be given the number of seconds since low time t. You need to say how many jellyfish you expect will be counted in 15 minutes and give a 95% confidence interval.

After thinking about it for a while, you figure that a pretty good model would be:

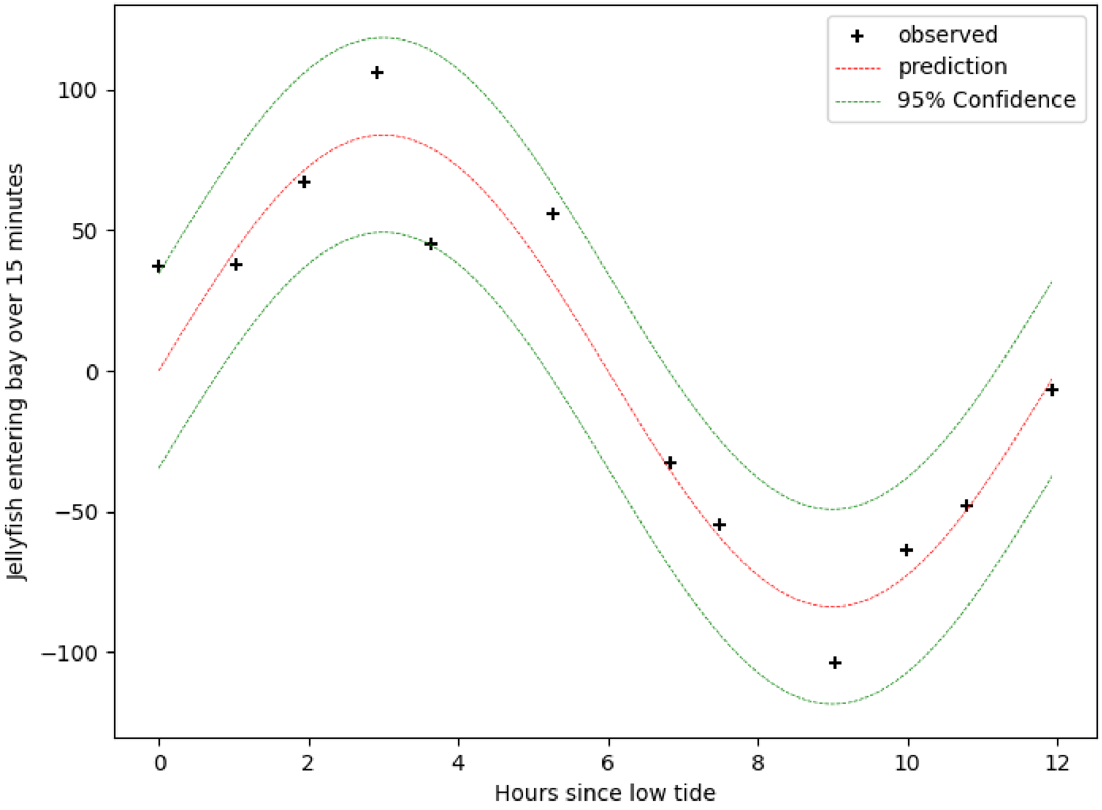
$$\hat{y} = m \sin(0.000145444t)$$

You need to use the data (and Bayesian modeling!) to find the best value of m, the expected maximum number of jellyfish going past the counter in 15 minutes. We will refer to this as ”magnitude” – as it is the magnitude of the sine wave.

(Take a moment to figure out why you are multiplying t times 0.000145444.)

Of course, reality won't match your expectation exactly. (Jellyfish do have propulsion and are not necessarily evenly distributed.) You decide to assume that the error will be normally distributed with a mean of zero. You need to use the data to find the best value of σ, the expected standard error.

After estimating these the magnitude and σ, you will produce a plot that shows the data you've been given, the curve representing your expected counts, and your 95% confidence intervals. Like this:



3 The Code

Create a program called `run_model.py` that reads in `samples.csv`. Convert the datetimes into the number of seconds since low tide. (Reminder: the first time is exactly low tide.)

Make an instance of `PyMC.Model`.

For priors, you can assume the PDF of the magnitude is uniform between 0 and 200. And you can assume the PDF of σ is a half normal distribution with $\sigma = 12$.

Create the `expected_count` using the magnitude, the times, and the function $PyMC.math.sin$:

$$\hat{y} = m \sin(0.000145444t)$$

Then you can express the model as

$$y = N(\hat{y}, \sigma)$$

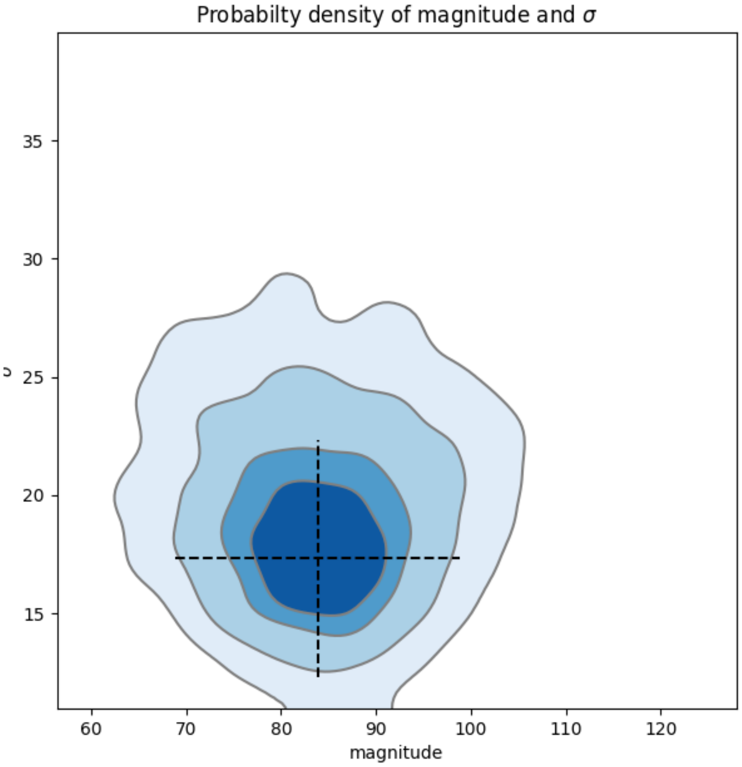
That is, reality will be a normally distributed around the expected value, with a standard deviation of σ. Supply the observed jellyfish counts.

Make chains of length 2000 after 500 samples of burn-in (or ”tuning”). (If this crashes, try setting `cores=1`.

Find the maximum a posteriori estimates for magnitude and σ. Print them out. It should look something like this:

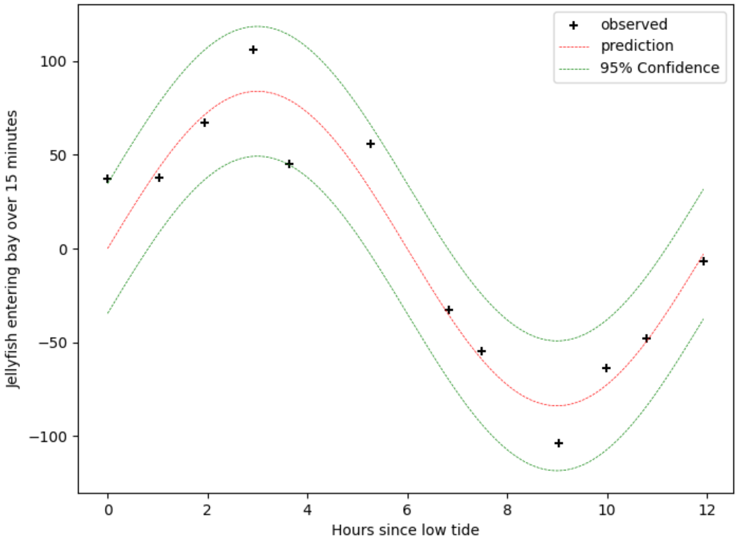
```
Based on these 12 measurements, the most likely explanation:
    When the current is moving fastest, 83.88 jellyfish enter the bay in 15 min.
    Expected residual? Normal with mean 0 and std of 17.31 jellyfish.
```

Get the posterior from the trace, and make a contour plot for the PDF of the magnitude and σ. Show the MAP values on the plot:



Save this plot as `pdf.png`.

Finally do a scatter plot of the data points you have and then add plots of the predicted values and the 95% confidence range based on the maximum a posteriori estimates



Save this plot as `jellyfish.png`

4 Criteria for success

If your name is Fred Jones, you will turn in a zip file called `HW13_Jones_Fred.zip` of a directory called `HW13_Jones_Fred`. It will contain:

- `run_model.py`
- `samples.csv`
- `pdf.png`
- `jellyfish.png`

Be sure to format your python code with black before you submit it.

We would run your code like this:

```
cd HW13_Jones_Fred
python3 run_model.py
```

Do this work by yourself. Stackoverflow is OK. A hint from another student is OK. Looking at another student's code is *not* OK.

The template files for the python programs have import statements. Do not use any frameworks not in those import statements.