Statistics Module

Homework 2.

Instructions: *This is an optional homework; turning it in or not will not affect your grade. If you do turn it in on time, we will grade it with feedback. Your answers must be handwritten; you can either turn in a hardcopy to Prof. Cowley's office (Freeman 113) or e-mail Prof. Cowley, the TAs, and Razan with a scan. Any use of resources is permitted.*

This homework is due by **Friday, September 26, 11:59pm**.

Problem 1.

[This is the same problem as in homework 2.]
You have K feature variables $\mathbf{x} \in \mathcal{R}^K$ with N samples as well as $y \in \mathcal{R}$ where $(\mathbf{x}_n, y_n)$ were collected together in pairs. You would like to train a ridge regression model's weights β with the best possible hyperparameter λ and then report your evaluated R^2 prediction performance.
Give pseudocode of this procedure and explain its steps.

Programming Problem 2.

You are testing various potential drugs to improve the white blood counts (WBCs) of human patients. For each test, you measure N control patients with leukopenia (a lack of WBCs) as well as M leukopenia patients which have taken the kth drug (where $k = 1, \dots, 1,000$). Each test was done independently. You want to identify the most promising drugs that increase WBC.
Download `hw3_prob2_control.npy` and `hw3_prob2_drug.npy` from `https://cowleygroup.cshl.edu/stats2025.html`.
Upload this dataset to Google colab:

```
wbc_control = np.load(hw3_prob2_control.npy)
wbc_drug = np.load(hw3_prob2_control.npy)
```

where `wbc_control` and `wbc_drug` each have shape $(K \times N)$ for the K drugs and N patients; note that some entries are not-a-number (NaNs) as some tests did not have the full N patients. These are the measured WBCs for the patients.

In Google colab, write a function to perform a permutation test of difference of means for a single test, the function takes as arguments the WBCs for control and drug populations and returns the p-value. Plot a histogram of the density versus p-value across all tests. Do you think some drugs worked?

You realize there are multiple hypotheses, and some drugs may seem to work by chance. Run the Bonferroni method to correct for multiple hypotheses with overall level $\alpha = 0.05$. How many drugs pass this method?

Now run the Benjamini-Hochberg method to control for the false discovery rate with overall level $\alpha = 0.05$. How many drugs pass this method? Plot p-value versus test index (sorted).

Programming Problem 3.

You present a batch of white noise images (grayscale) and record a neuron's responses from mouse visual cortex. You want to identify the 'receptive field' of the neuron by regressing the image pixels onto responses.
Download `hw3_prob3_images.npy` and `hw3_prob3_responses.npy` from `https://cowleygroup.cshl.edu/stats2025.html`.
`images`: (num_images, num_pixels, num_pixels) with pixel intensities between 0 and 1
`responses`: (num_images,)
Separate the data into a training set (80%), a validation set (10%), and a test set (10%). Perform ridge regression (hint: use sklearn's package `Ridge` and reshape images into vectors) by choosing an appropriate value for hyperparameter λ. Report the Pearson correlation squared (hint: use `np.corrcoef`) versus lambda evaluated on both the validation data and the test data (two curves). In addition, plot the recovered β weights in the form of an image...What is the neuron's receptive field? [Hint: With matplotlib, plt.imshow(beta, cmap='gray').]

Programming Problem 4.

You are recording from an electrode in the brain and save each waveform that crosses a voltage threshold. Some of these waveforms are likely noise, and some of these waveforms are actual action potentials or "spikes"—this is called spike sorting. We want to identify which waveforms belong to which neurons in an unsupervised way. Each waveform lasts 30 ms (30 time points), and you realize you can treat each time point as a "feature variable".
Download `hw3_prob4_waveforms.npy` from `https://cowleygroup.cshl.edu/stats2025.html`.
`waveforms`: (num_waveforms, num_timepoints)
Apply PCA to the waveforms [Hint: Use sklearn's PCA], where the time points are the feature variables.
Plot the explained variance versus PC index identified by PCA. How many neurons N do you identify?

Plot the loadings of the first N PCs.

Code up an algorithm to label each waveform as coming from one of N neurons (or 'none').