

Coding Homework #2

Instructions:

1. Make a *Markdown Cell* that contains the following information:
 - Your name (small header)
 - The name of this class (italicized)
 - The date (bolded)
2. Make a *Code Cell* that imports `numpy` as `np`, `matplotlib.pyplot` as `plt`, `statistics` as `st`.
3. Make a *Code Cell* that imports your `v_rest_sst.csv` and `v_rest_pvalb.csv` files as `numpy` arrays called `v_rest1` and `v_rest2` respectively. You're used to doing this with the `pd.read_csv` function for `pandas`, but can accomplish the same thing using the `np.loadtxt` function for `numpy`.
4. Make a *Code Cell* that concatenates `v_rest_sst` and `v_rest_pvalb` into a matrix with 2 rows and 13 columns named `v_rest`.
5. Make a *Code Cell* that prints the `size` of `v_rest`.
6. Make a *Code Cell* that returns the first row of `v_rest`.
7. Make a *Code Cell* that returns the sum of the values in the third column of `v_rest`. You can accomplish this by using `v_rest[:,2].sum()`, with your index inside of the brackets. As an example, if I wanted to return the sum of the first row of `v_rest`, I could use `v_rest[0].sum()` to get my answer.
8. Make a *Code Cell* that returns the second-to-last value in the second row of `v_rest` (use negative indexing for the column value).
9. Create a *Code Cell* that makes a histogram of the values in the first row of `v_rest` (you can do this by using the `plt.hist()` function). Label your axes.
10. Create a *Code Cell* that takes the mean of the values in the first row of `v_rest`, and assigns the mean to a variable called `v_rest_mean_sst`. Do the same thing for the second row of `v_rest`, and assign the mean value of the second row to a variable called `v_rest_mean_pvalb`. You can get the means by using the `st.mean()` function (remember to put what you want the mean of inside the parentheses of the function).

11. Create a *Markdown Cell* that describes the difference in means between the two rows. Which one is bigger?
12. How can we test whether the difference in means is *meaningful* (statistically significant)? When we're comparing two means, we can use a t-test (assuming the data are normally distributed). To use a t-test on our data, we will first need to import the `scipy` package. Create a new *Code Cell* that runs the following:

```
import scipy.stats as stats
```

13. Now, we can perform a t-test on the two rows of `v_rest`. To do this, create a new *Code Cell* that uses the `stats.ttest_ind()` function - for example:

```
stats.ttest_ind(row1, row2)
```

Replace `row1` and `row2` in this case with the first and second rows, respectively, of `v_rest`.

14. Make a *Markdown Cell* that describes the results of your t-test. You should have `statistic` variable and a `pvalue` variable. The way to interpret your p-value is as a probability - specifically, the probability that the mean of one distribution could be found in the second distribution. Typically, results are considered to be "statistically significant" if the probability is lower than 5% - or 0.05. What is your p-value? Based on this p-value, do you think it is likely that the means are significantly different, or do you think it is likely that they are not?
15. Make sure to add at least one line with a `#` in front of it to each *Code Cell* to describe what each cell does.
16. Save a copy of this notebook to your GitHub repo with the title "coding_homework_2.ipynb".

