Name: _____

Date: _____

Phys 194–FYRE Assignment #4
Basic Statistics, Plotting & Fitting a Model to Data

Assignment Policy: You can consult class notes, books, and online resources. You can work in small groups (2 or 3), but you must turn in your own work. Make sure you are clear about the process you use to solve the problems: partial credit will be awarded.

Getting Started

For today's activity, we will again make use of the NANOGrav notebooks to practice coding in python. Login with your designated account number `uwm.researchNN` and navigate to your own directory as you've done before. In order to complete this assignment, you will need to make use of two files available in the course's directory, `dm-dist.txt` and `gb.txt`, so copy these into your own directory.

Make a new python notebook called `Assignment4` in your directory, change the first cell's type to *markdown* and record your names and date. Make another cell (also markdown-type) and label it with the heading of this section. Do this with each section of this assignment to make your notebook better organized and easier to read. In markdown-speak, a single '#' character placed directly before the text formats your text as a header, '##' denotes a sub-header, and with no preceding symbols, your text will appear as normal. Remember, you can run individual cells with `shift + return`.

Finally, we will be making extensive use of both `numpy` and `pyplot` libraries today, so you may want to simply copy the full cell of import commands from `Assignment3.ipynb` to get started quickly. As you go through each exercise, remember to include relevant code in your notebook *and* answer questions in this packet.

Write Basic Statistics Functions

Load data into a python array from `gb.txt` using the command `gb = np.loadtxt('gb.txt', dtype='float')`. This dataset includes Galacitc latitudes for all known pulsars – their angular distance (in degrees) from the Galactic plane.

1. To convince yourself that the dataset is complete, try running `wc` on the text file in a terminal or use the `len()` function in your notebook. How many measurements are in the file?

2. Write your own code that loops over `gb` to compute **mean** and **standard deviation** statistics for the full dataset. What values do you find? You may want to refer back to class notes for the summation expressions.

3. Write a *function* to compute the **median**. What is it?

4. Use google to look for corresponding functions included in numpy. What are they and how do the results compare to yours?

Plotting a Distribution

Remember to make a new header in your python notebook for this exercise. We will continue to use the `gb` array.

1. Use the `plt.hist()` function to plot a histogram, showing the distribution of Galactic latitudes. Experiment with different numbers of bins in your histogram; what changes do you notice? If you're unsure about how to manipulate the histogram's settings, google it! Try looking for some examples.

2. Does the distribution look as you would expect it to? Do you notice any interesting features? Convince yourself that the statistics you computed before make sense by looking at the distribution.

3. Does it look like a Gaussian function would be a good fit to the distribution? Why or why not?

Fit a Line to Data

Load two columns of data from `dm-dist.txt`, representing dispersion measures ($\mathrm{pc\,cm^{-3}}$) and estimated distances (kpc). I suggest you use the `unpack=True` keyword parameter to speed things up, like so:

```
dm, dist = np.loadtxt('dm-dist.txt', dtype = 'float', unpack=True)
```

This dataset is made up of pulsars with Galactic latitudes, $b < 5°$, but only those in the direction opposite Galactic center (e.g. Galactic latitudes, $90° < \ell < 270°$).

1. Refer back to class notes and write code that uses least squares to find the line of best fit for the data given, where x_i and y_i represent DM and estimated distance, respectively. For simplicity, feel free to assume the y-intercept, $b = 0$. What is the slope of the line? For a pulsar with a DM of $30\,\mathrm{pc\,cm^{-3}}$, what is a reasonable distance estimate according to your model?

   Use the `plt.scatter()` function to plot DMs versus distances. Over-plot your best-fit line. Do you think your fit is a good representation of the data? How might you improve your fit?