For numerical problems, you must show your work to receive credit.

Problem Set 1

1. (20 points) A weighted six-sided die is weighted such that throwing a 1 or 6 is three times as likely as throwing any other number. Compute the mean and variance of the die: E[X] and $E[(X-\mu)^2]$. How do these differ from mean and variance of a fair die and why?
2. (20 points) Suppose two class conditional probability distributions can be represented by normal distributions: $P(x \mid \omega_A) \sim n(\mu_A, \sigma_A^2)$ and $P(x \mid \omega_B) \sim n(\mu_B, \sigma_B^2)$. Given prior probabilities $P(\omega_A) = ¾$ and $P(\omega_B) = ¼$, compute the decision threshold for Bayes' decision rule[1].
3. (80 points) Write a Python program that meets the following criteria.
   - All classes and functions other than your driver function must be in package mydsp (see package tutorial, basically you create subdirectory mydsp, and create an empty __init__.py file.

     As modules are imported by filename, part of the interface includes the filename. You should have the following filename structure. Please note that your code will be tested on a linux system and that file case is important:

     top-level directory:
     i. mydsp
        1. __init__.py
        2. audioframes.py (contains AudioFrames)
        3. rmsstream.py (contains RMSStream)
        4. plots.py (contains fit_norm and plot_data)
     ii. driver.py
   - Code must be well commented.
   - The following classes and functions should be written to have the following signatures and behavior:

```
class AudioFrames:
    """AudioFrames
    A class for iterating over frames of audio data
    """

    def __init__(self, filename, adv_ms, len_ms):
```

[1] Recall from high school that the quadratic equation, $\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$, finds the roots for polynomial $ax^2 + bx + c$. You'll need to think about which root is appropriate.

```
        """AudioFrames(filename, adv_ms, len_ms)
        Create a frame generator from file filename where each is
        len_ms milliseconds long and frames are advanced by adv_ms.
        """

    def get_framelen_samples(self):
        "get_framelen_samples() - Return frame length in samples"

    def get_framelen_ms(self):
        "get_framelen_ms() - Return frame length in ms"

    def get_frameadv_samples(self):
        "get_frameadv_samples() - Return frame advance in samples"

    def get_frameadv_ms(self):
        "get_frameadv_ms - Return frame advance in ms"

    def get_Fs(self):
        "get_Fs() - Return sample rate"

    def __iter__(self):
        "__iter__() - return iterator"
        return self

    def __next__(self):
        "__next__() - return next frame"
```

Use Scientific Python's scipy.io.wavfile package to read the data. Note that this function returns a numpy array which is an efficient representation of data with a large number of available operations.

If you are unable to import scipy.io.wavfile, you have a problem with your installation. See the instructions for installing Anaconda and additional required packages.

```
#module mydsp.rmsstream
class RMSStream:

    def __init__(self, frame_stream):
        """RMSStream(frame_stream) - Initialize a root mean square stream
        from an instance of AudioFrames.
        """

    def __iter__(self):
        "__iter__() - return iterator"
```

```
    def __next__(self):
        "next - next RMS value"
```

and the functions in module mydsp.plots:

```
def fit_norm(data):
    "fit_norm(data) – Return tuple with estimator of mean and variance: (μ, σ^2).
```

The numpy functions numpy.mean and numpy.var will be helpful here.

```
def plot_data(data, n1, n2)
    """plot_data(data, n1, n2)
    Plot a histogram of specified data.  Overlay the plot with two normal distributions.
    n1 and n2 are tuples describing the normal distributions' mean and variance (e.g. the
    output of fit_norm()).
    """
```

Matplotlib should be used to produce the plot. A general tutorial can be found here and describes how to produce simple plots including histograms. A more tricky example is how to use more than one scale on the same plot. If you have problems with functions expecting numpy arrays, you can convert a Python list of numbers to a numpy array using the numpy.array function.

The assignment contains an audio file that you should download to your machine. Write a driver program that meets the following criteria:

- Driver is in file driver.py and is in the main directory (sibling of subdirectory mydsp).
- Constructs an AudioFrames object for the file with a 10 ms advance and 20 ms length.
- Constructs and RMS object using the AudioFrames object.
- Builds a list of RMS values.
- Estimate the mean and variance of the speech and noise distributions:
  - speech – Use training data between 1.3 and 2.8 s.
  - noise – Use training data between 0 and 1.3 s.
- Create the following plots:
  - RMS signal with a time axis.
  - A histogram of RMS data overlaid with noise and speech distributions (mydsp.plots.plot_data). Optionally you can draw the Bayes' decision boundary (no extra points, just the satisfaction of a job well done).

Save the plots and insert them into the hard copy that you turn in. Make sure that each plot has the axes labeled and write an informative caption. In scientific papers, a good caption provides enough information for a reader familiar with the

techniques to typically understand the plot without having to read the paper. As an example, consider the figure and caption written by Van Cise et al. (2017):

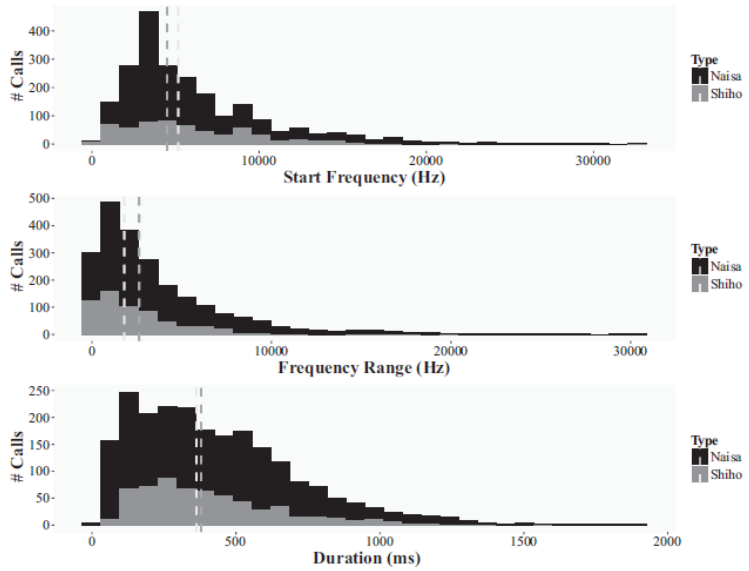
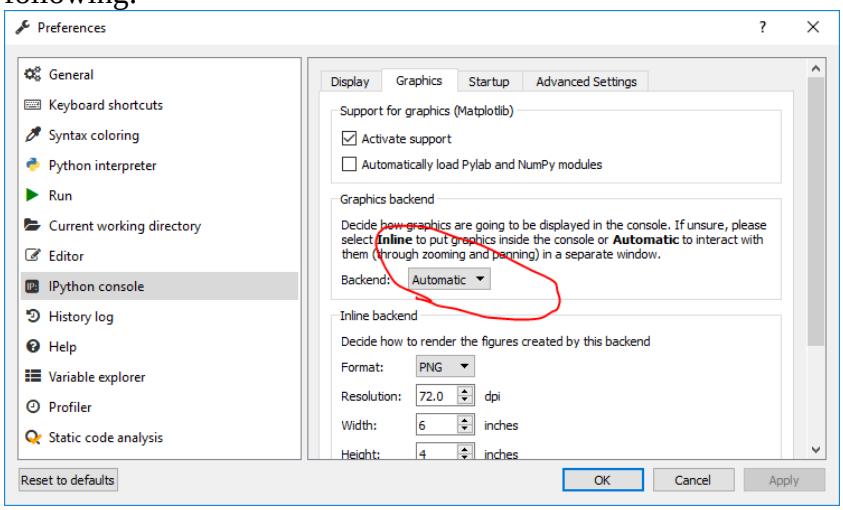



FIG. 7. Histograms of start frequency, frequency range, and duration of calls from Naisa- and Shiho-type short-finned pilot whales. Dashed lines represent median values for both types.

A couple things to remember:

- If you are not using ipython (e.g. running Spyder from Anaconda), plots will not show up as soon as you execute them.
- On Spyder's ipython, plots appear in the log window and are tiny. To make them appear as separate interactive windows, select Tools/Preferences and set the following:




What to turn in:

- In addition to the written problems, turn in
  - a print out of your code.
  - output of a run
  - the requested figures with captions

- Submit to blackboard a zip or tar archive of your code. The top level of the archive should contain your driver program and the mydsp module. As a reminder, unit tests will be used to check functionality. If you do not meet the specified interface, your code will not function.

Works Cited

**Van Cise, A. M., Roch, M. A., Baird, R. W., Aran Mooney, T. and Barlow, J.** (2017). Acoustic differentiation of Shiho- and Naisa-type short-finned pilot whales in the Pacific Ocean. *J Acoust Soc Am* **141**, 737-748, doi:10.1121/1.4974858.