# Homework 2: Dealing with Data

October 5, 2014

## 1 Getting Started

For this homework you will be working with data input from external data files and users. In astronomy you will often find yourself having to incorporate the data of different research groups, and so it is useful to learn how to incorporate external files.
If you are having issues getting started I would reccomend writing this program using modular programming with a main procedure used to call all your smaller worker functions and procedures as needed. If you are **REALLY** stuck then I have included another file in the Week6-Plotting/homework2 repo called *hw2skeleton.pro*. This should only be used as a crutch after you've attempted to solve whatever step of this homework you're stuck on.

## 2 Reading in Data

To begin you will need to write a function that will read in the three columns from a data file into three different varibles using readcol. **Don't look at the first two columns!!!** It will be more amusing at the end if you don't. Depending on your last name you will read in a different file. See Table 1 for what file you should read in.

To read in the data it will be necessary to use special options for the readcol command, 'format' and 'delim'.
Format defines what data type the data will be in the form of a string of letters separated by commas for each column. To read in a string column, then a float column, then an integer column, the option would be: format='A,F,I.'
The delim command provides a user-defined delimiter to identify how the

| First Letter of Last Name | File to Load In |
|---|---|
| A-G | one.txt |
| H-M | two.txt |
| N-Z | three.txt |

Table 1: Table of what file to read in based off your last name.

data is divided between columns. If I had a file with the top line, 'red fish blue fish one fish two fish dog', I could provide the option, delim = 'fish' to readcol, and it would read in the first values for the three columns to be red, blue and fish.
After you read in the data, load it into three columns of a three by n-elements(col1) and return that array.

# 3 Input from User

Next, write a function that takes in the third column of data from the file and iteratively asks the user for input based on the values of the third column. To do this you will need to use the read command. As an example, to ask for the first value of the third column (which I will have given as input to the function in a variable I call values) and save it to an array of user answers, I would write:

```
response = ''
read, "Please provide a "+values[0], response
answers[0] = response
```

It would be useful to use a for loop or while loop to iteratively ask these questions (if you type each one out I will cry). Once you have these answers, I would also like you to save them using the save command so that they can be looked at later (Bonus points for making us laugh). To do this, you'll want to write a command like this:

```
save, answers, filename = 'answers.sav'
```

Have the function then return the array with all the user answers.

## 4 Mix and Mash Input and Data

Now that you have your user input, you want to use it with the data you read in. Write a function that will take in the first and second columns of the data you read in, along with your user generated answers, and create a three by n-elements([data column]) array. Have the first column be the first column of data, the second column be the array of user answers and the third column be the second column of data. Have your function return this array.

## 5 Output

Write a procedure that will take the array from step 3 (I'll call it arr3) and read out each row iteratively. To read row 23, for example, I might use the command:

```
print, arr3[*,22]
```

or

```
print, arr3[0,22], arr3[1,22], arr3[2,22]
```

The second option will space things more neatly.

## 6 Main Procedure (Almost done!)

Finally, write a procedure that runs all of the 4 above functions and procedures, in order. (Remember to compile twice when you run everything!) Make sure everything runs fine and then save your .pro file to your personally repository and push everything back up!