**Python Lab 9a Getting Started with Data Science**

A. The labs in this chapter will help you get started processing data in are starting to discuss Data Science.

B. Open a Python program in your normal development environment (if you were using CodeSkulptor t).

C. Try out the following code. It makes a Pie Chart. Can you find where the pie chart is saved? Open it.

|  |
| --- |
| **import matplotlib.pyplot as plot**  **# set up your lists**  **numlist = [8, 6, 5, 3]**  **namelist = ['freshmen', 'sophomores', 'juniors', 'seniors']**  **colorlist = ['red', 'green', 'pink', 'yellow' ]**  **explodelist = [0.1, 0.0, 0.0, 0.0]**  **# make the pie chart**  **plot.pie(numlist, labels=namelist, autopct='%.2f%%', colors=colorlist,**  **explode = explodelist, startangle = 90)**  **plot.axis('equal')**  **plot.savefig('piechart.png')** |

D. Once you can find the saved pie chart, go back into the program and make changes. In particular, make sure you understand how to:

- match up name with numbers and colors

- use the explodelist features

- change the start angle

E. Search on the Web to see how the autopct command works. Try different values to print out no decimal place, then one decimal place.

F. Search the Web to learn how to use subsetting in Python lists. Practice making subsets of the namelist.

**Python Lab 9b Histograms**

A. Make a new program and add these lines:

**import matplotlib.pyplot as plot**

**numbers = [2, 2, 7, 5, 2, 6, 9, 1, ……, 8] # lots of numbers, all on one line**

**print (numbers)**

**plot.hist(numbers, bins=3, color = <some color>, alpha = 0.8)**

**plot.savefig(‘hist1.png’)**

B. Make sure you can open the .png file you saved.

C. Change the number of bins, change the color, run the program again, and look at the new image.

D. Make a second list, and a second histogram with a different color. Plot it. Look at your png.

E. Challenge: Write code that sets the number of bins to one third of the number of items in each list.

F. Take a look at this Google Doc: <http://tinyurl.com/lhn567g>

|  |
| --- |
| This data is a list of lists !! In your python file, copy the contents of the Google Doc.  **female\_country\_heights = [**  **[‘Afghanistan’, 1.638],**  **[‘Algeria’, 1.62],**  **<many more rows>**  **]** |

G. Now, let’s print out some information to see if we really have the data in that list.

Print out the length of your list. Print out the first row of your list

H. Now, grab just the heights and put them into another list:

**female\_heights = \_\_\_\_\_\_ # empty list**

**for each\_row in female\_country\_heights:**

**print (each\_row)**

**height = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ # just the height from each row**

**print(height) # print it**

**female\_heights.append(height) # append this height to our new list**

**print \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ #length of female\_heights**

**print \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ # first row of female\_heights**

I. Now, make a histogram of the heights.

**plot.hist(female\_heights, bins=7, alpha = 0.5, label = 'female', color = 'blue')**

**plot.savefig(‘hist2.png’)**

**Python Lab 9c Charts from CSV Files**

A. Students will collect data. The instructor will create a Google Sheet with a name like “day3” and will make the Google Sheet available to the students to edit. Students will write their answers in the document.

1.) How many hours of sleep did you get last night? (as a float)

2.) What is your height in inches? (as a float)

3.) According to Google Maps, what is the distance in miles from your house to school? (as a float)

B. Wait until all students have entered their data. Then open the file and do the following:

1. On your own computer, choose “download as CSV” and save to your computer.

2. Upload the file to your Python account.

3. Make a Python file called day3.py and enter this code:

**import \_\_\_\_\_\_\_\_\_\_\_ as np**

**import \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.pyplot as plot**

**datatable = np.genfromtxt(**

**'day3.csv', # or whatever your file is named**

**skip\_header=1, # skip the first line**

**delimiter= ',' , # this is a Comma Separated Value file**

**dtype='float', # data type is float**

**filling\_values=0, # in case a data element is missing**

**usecols = (0,1,2)**

**)**

**print (datatable)**

**print ('this table has ' , len(datatable), 'rows')**

4. Using subsetting techniques, (do a search if you don't know how) create three separate arrays,

one for the hours of sleep sleephours = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

one for the heights heights = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

one for the distances distances = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Print out each array separately.

C. Make Histogram of distances. Use your knowledge of making histograms from DataCamp.

Save it as “day3\_hist.png.” Include an x-label and a title.

Show it to your teacher.

D. The plot.clf() function resets your plot. You need to use this so that the two plots do not show up together.

Make a scattergram of the two variables hours of sleep and height in inches. Include an x-label, a y-label, and a title. Save it as “day3\_scatter.png.”