04-2 Graphics and Plots Extended Example

CSI 500

Example: analysis of undergraduate student height

- Let's use our Python plotting skills for a small example data set
  - We want to create some line plots
  - We want to create some box and whisker plots
- Our data is a CSV file that looks like this
  - A copy of 'student_data.csv' will be supplied with the homework #4 assignment…

```
gender, height, year
female, 56, freshman
male, 63, freshman
male, 81, sophomore
female, 71, freshman
male, 83, senior
female, 63, senior
female, 68, junior
male, 60, freshman
male, 81, freshman
```

Get the data

- Read the file (we can ignore the first line with column headers)
- store the data as a List of 3-element tuples
    - (gender, height, year)

```
#
# let's read in the data
#
infile = open('student_data.csv', mode='r')
header = infile.readline()
file_data = infile.readlines()

student_data = []
for item in file_data:
   item = item.strip()
   gender, height, year = item.split(',')

   gender = gender.strip()
   height = int(height)
   year = year.strip()

   student_data.append( (gender, height, year ))
```

Now process the data

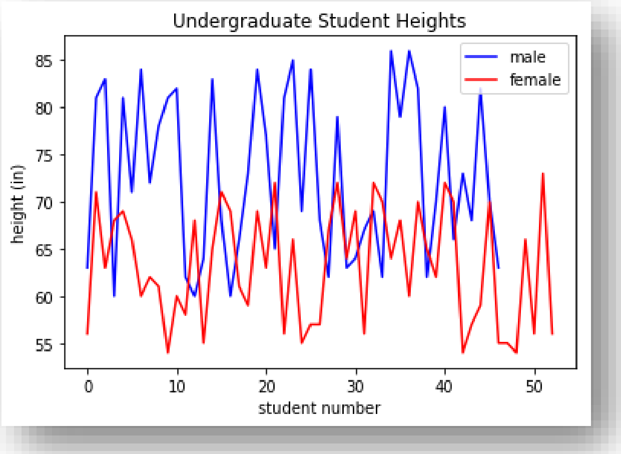
- Let's build a (nonsensical) plot of student heights by gender
- make Lists to store male and female height data
- run thru the data set, extracting male and female height data
- save height data in appropriate data List

```
# let's plot the observed heights by gender
#
male_heights = []
female_heights = []

for item in student_data:

   gender = item[0]
   height = item[1]
   year = item[2]

   if gender == 'male':
      male_heights.append( height )
   else:
      female_heights.append( height )
```

Now make the x-y plot

- call plot() with each set of data we want to plot
- use plot parameters to specify color and data label ('-b' means blue, '-r' means red)
- specify title, xlabel and ylabel
- specify default legend
- display the plot using plt.show()

```
plt.plot(range(len(male_heights)), \
            male_heights, '-b', label='male')

plt.plot(range(len(female_heights)), \
            female_heights, '-r', label='female')

plt.title('Undergraduate Student Heights')
plt.xlabel('student number')
plt.ylabel('height (in)')

plt.legend()
plt.show()
```



Now process the data again

- Let's build a (somewhat less nonsensical) box plot of student heights by gender and class year
- make a bunch of separate Lists to store male and female heights by class year

```
#
# now lets do a boxplot by gender
# and class year
#
male_freshman = []
male_sophomore = []
male_junior = []
male_senior = []
female_freshman = []
female_sophomore = []
female_junior = []
female_senior = []
```

Now make the x-y plot

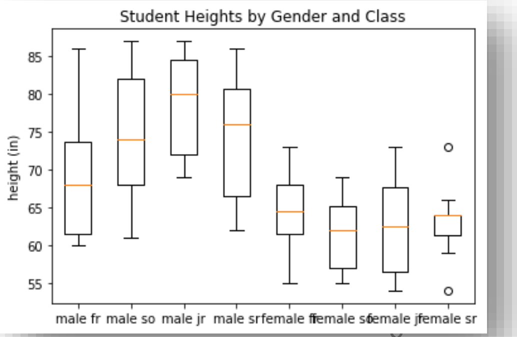
- Iterate over the data set, and filter out the records one at a time
- extract gender, height, and year
- test for gender and year, and put data into the appropriate storage List

```
for item in student_data:
    gender = item[0]
    height = item[1]
    year = item[2]

    if gender == 'male' and year == 'freshman':
        male_freshman.append( height )
    if gender == 'male' and year == 'sophomore':
        male_sophomore.append( height )
    if gender == 'male' and year == 'junior':
        male_junior.append( height )
    if gender == 'male' and year == 'senior':
        male_senior.append( height )

    if gender == 'female' and year == 'freshman':
        female_freshman.append( height )
    if gender == 'female' and year == 'sophomore':
        female_sophomore.append( height )
    if gender == 'female' and year == 'junior':
        female_junior.append( height )
    if gender == 'female' and year == 'senior':
        female_senior.append( height )
```

Now make the box plot

- make a combined data set with all the data that we want to plot
- use the labels option to specify a label for each boxplot
- specify title
- specify ylabel
- display the plot using plt.show()

```
#
# now do a box plot with
# multiple data sets as inputs
#
combined_data = [male_freshman, male_sophomore,
                 male_junior, male_senior,
                 female_freshman, female_sophomore,
                 female_junior, female_senior]

plt.boxplot(combined_data,
            labels=['male fr', 'male so',
                    'male jr', 'male sr',
                    'female fr', 'female so',
                    'female jr', 'female sr'])
plt.title('Student Heights by Gender and Class')
plt.ylabel('height (in)')
plt.show()
```



Summary

- Use plot() for basic 2-D X-Y plots
  - multiple lines can be plotted on same graph
  - use colors and styles to distinguish
  - use legends and labels to improve readability
- Use boxplot() for box plots
  - multiple data sets can be plotted on same boxplot
  - use y-axis legend and x-axis labels to improve readability