

Plotting Graphs for ECE 358

Bernie Roehl, Fall 2020

For the labs in this course, you are expected to produce graphs. You are free to do this however you choose. For example, if you use Python, there's the matplotlib library. There are similar libraries for other languages.

However, since generating graphs is not the main focus of the labs, we're providing the following simple instructions that will allow you to easily produce graphs using the standard Linux `gnuplot` command.

Generating Your Data

As your program runs, you'll generate a series of data points. For example, in Lab 1 Question 3 you'll be producing a value of $E[n]$ for each value of ρ . Write those values out to a file. The format is simply the X and Y values (in this case ρ and $E[n]$) separated by a space, with each pair of values on its own line.

Using Gnuplot

Create a text file containing the following:

```
set title "Average Number of Packets in Queue vs Utilization"
set xlabel "Utilization"
set ylabel "Average Number of Packets in Queue"
set xrange [0:1]
set yrange [0:6]

# no "key" or "legend" required
unset key

# use a png file for output
set term png
set output "q3_graph1.png"

# plot lines and points, with solid dots as the markers
plot 'q3_data1' with linespoint pointtype 7
```

The contents of this file should be self-explanatory. Assuming that this text file is called `q3_graph1`, you can just give the command `gnuplot q3_graph1`. You can add that command to your Makefile or shell script.

Running that command will produce a PNG file called `q3_graph1.png` in your current directory. You can display the PNG file by typing `display q3_graph1.png`. And of course, you can also put the PNG file in your report.

Multiple Curves

The procedure is the same for graphs that require multiple curves (as is the case in Lab 1 Question 6, for example).

The data you generate will be in multiple “sets”. For example, you’ll have a set of data points for the $k=10$ case, another for the $k=25$ case and another for the $k=50$ case. Simply write those sets of data to the file, and put two blank lines between each set.

Your gnuplot file for the second graph in question 6 would look like this:

```
set title "Packet Loss vs Utilization"
set xlabel "Utilization"
set ylabel "Probability of packet loss"
set xrange [0:1.5]
set yrange [0:0.5]

# position the 'key' or 'legend'
set key center right
set key spacing 2

# output to a png file
set term png
set output "q6_graph2.png"

# "index" refers to which dataset in the file to plot
# Note that we use different markers for each dataset
plot 'q6_ploss_data' index 0 title "k=10" with linespoint pointtype 5, \
      'q6_ploss_data' index 1 title "k=25" with linespoint pointtype 6, \
      'q6_ploss_data' index 2 title "k=50" with linespoint pointtype 7
```

Note that the `\` character at the end of a line indicates continuation, i.e. the next line in the input file is actually part of the same command. So the last three lines above are actually a single plot command, and you could simply have it as one long line without the `\` characters (assuming readability isn't a concern).

And that's all there is to it. From these two examples, you can easily figure out how to do the remaining graphs. **Hint:** Change the labels and ranges of the graphs, and the names of the input and output files.

Again, you don't *have* to do it this way. It's up to you how you generate your graphs, as long as it's very straightforward for the TAs to repeat the procedure.