Pattern Recognition Homework Guidelines

The most basic skill in this class is *analysis*: can you interpret the information to answer the question. The code may run without crashing, but are we looking at an answer or gibberish? So we also need to address the deeper problem: *do you know what you are doing*?

Your job is produce a notebook containing both your code and your analysis of the results. Think of this as a report you could give to your boss, client, or team members to describe your work. Tell enough to get across that you know what you are talking about and that **you have adequately solved the problem**.

Include the following:

- Header – Name, date, class, assignment, and one-sentence summary
- Problem description
- Solution Method - your plan of attack. Explain what you are going do to solve the stated problem.
- Input – section of code where you load and pre-process data
- Analysis – analyze the data
- Results – plots and/or data tables showing solution to the problem
- Discussion – a paragraph or two justifying design decisions you made and methods you chose, describing problems you encountered, how the solution could be improved, and lessons learned

Points	**0**	**1**	**2**
Problem Description	None	Description of the problem is lacking	The problem is explained and motivated well
Solution Method	Shows no helpful plan for what you actually did to solve the problem	Incomplete, confusing and/or the method does not fit the problem	Demonstrates clear understanding of what to do and why it is appropriate for the problem
Analysis	Mostly incomplete or inappropriate	Lacking, partially complete, or difficult to follow	Clear and easy to follow
Results	Results are not given, or are irrelevant to problem	Results unclear or poorly formatted; too little or too much is shown	Clear presentation showing solution to the problem
Discussion	Does not demonstrate that you knew what you were doing	Shows partial understanding, poor design, and/or misinterpreting results	Proves clear understanding of problem and solution, shows careful design, draws proper conclusions

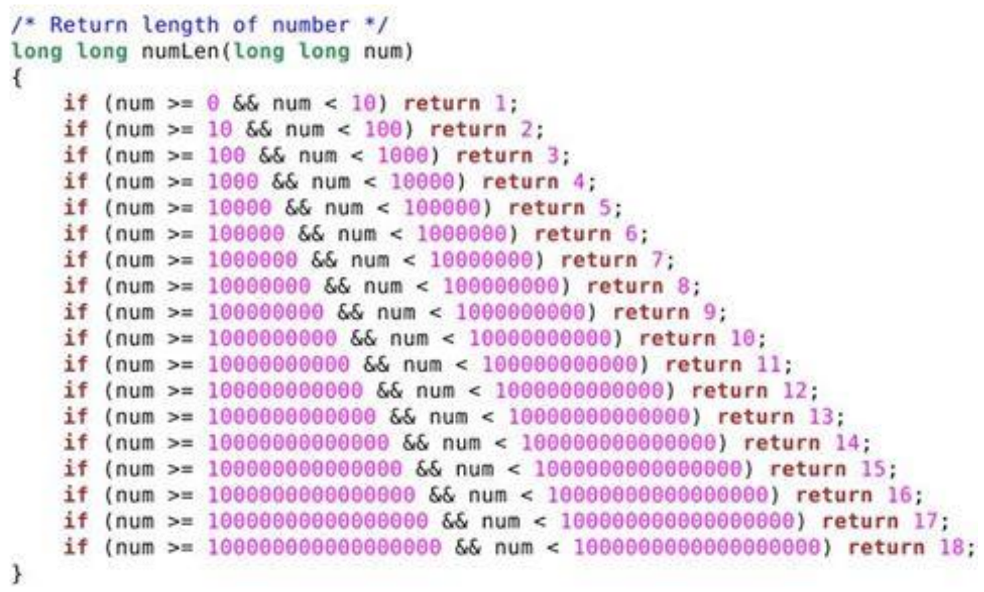
Other Comments

Readability

- http://abstrusegoose.com/432
- Is it possible for someone else to look at your code and know what's going on?
- Note that good comments are not the same as long comments! Having comments that are too long, unhelpful, and numerous is just as unreadable as having no comments.
- Variable names can count as comments:
  - This is ok: `force = mass * accel;`
  - This is not ok: `force = mass * accel;  % Apply Newton's first law to calculate force=mass*acceleration`
- Too many results/plots/tables is just as bad as not enough! More is not always better. Don't make us slog through a core dump of mostly irrelevant stuff to get to the answer.

Implementation and Mastery

- There is a difference between good code and working code



```
/* Return length of number */
long long numLen(long long num)
{
    if (num >= 0 && num < 10) return 1;
    if (num >= 10 && num < 100) return 2;
    if (num >= 100 && num < 1000) return 3;
    if (num >= 1000 && num < 10000) return 4;
    if (num >= 10000 && num < 100000) return 5;
    if (num >= 100000 && num < 1000000) return 6;
    if (num >= 1000000 && num < 10000000) return 7;
    if (num >= 10000000 && num < 100000000) return 8;
    if (num >= 100000000 && num < 1000000000) return 9;
    if (num >= 1000000000 && num < 10000000000) return 10;
    if (num >= 10000000000 && num < 100000000000) return 11;
    if (num >= 100000000000 && num < 1000000000000) return 12;
    if (num >= 1000000000000 && num < 10000000000000) return 13;
    if (num >= 10000000000000 && num < 100000000000000) return 14;
    if (num >= 100000000000000 && num < 1000000000000000) return 15;
    if (num >= 1000000000000000 && num < 10000000000000000) return 16;
    if (num >= 10000000000000000 && num < 100000000000000000) return 17;
    if (num >= 100000000000000000 && num < 1000000000000000000) return 18;
}
```

- Really, really ugly solutions that work are still ugly. Did you take 20 lines of code to do something that could have been done in 2?
- Do you know the language well enough to take advantage of built-in features instead of doing things the hard way?
- That being said, code that is too clever is also hard to read. Don't try to cram in one line of dense gibberish if two lines is easier to understand.

Solution

- Only 20% of your grade comes from getting the “right” answer (like there is such a thing) in the right way. If this is all you do, you will do poorly.
- Answer all parts of the problem

Output Formatting

- Make sure every plot has title, labels, and is correctly formatted.
- Don’t just display a bunch of numbers without explaining what they are. Make lovely tables and/or tell us what these numbers mean

Grammar and Writing

- y’all it important b write good
- Technical writing should be clear, concise, and easy to read.