PHYS 210, Assignment 7

Create a new directory somewhere in your home directory with the name `yourusername_assignment_7` to store the files you will create for this assignment. To hand the assignment in, copy the directory with your results to `/home2/phys210/yourusername/`. Make sure it's there and has the right permissions (read and execute for everyone, write for you).

1 Loops

1. Solve exercise 1 from chapter 7 (`http://greenteapress.com/thinkpython2/html/thinkpython2008.html`).
2. Solve exercise 4 from chapter 8 (`http://greenteapress.com/thinkpython2/html/thinkpython2008.html`).
3. In exercise 2 from assignment 5 (Functions II), you ended up calculating the mean, variance, and standard deviation for a single column. Repeat that exercise, but now calculate the mean and variance for all columns.

Put the commented code in a file called `loops.py`.

2 Monte Carlo I

Many integrals that you will encounter in your physics career do not have analytical solutions. The only way to solve them is using numerical methods. Computing certain integrals, especially with many dimensions, can be very hard and time consuming, even with numerical integration methods, such as quadrature. A surprisingly simple but effective methods is Monte Carlo integration (`https://en.wikipedia.org/wiki/Monte_Carlo_integration`).

As a very simple example of Monte Carlo integration, you will calculate the area of a unit circle ($A = \pi$) in this exercise. The algorithm works as follows:

1. Create N random points on the unit square, as show in figure 1.
2. For each point, check if it lies within the unit circle. Count the number of points within the unit circle
3. The ratio of the number of points within the unit circle and the total number of points will approach the ratio of the area of a quadrant of the unit circle and the area of the unit square, i.e., $\frac{\pi}{4}$.

Implement this algorithm to calculate the area of the unit circle. One possible approach is to create two arrays x and y holding N random numbers between 0 and 1 and then loop over all pairs to check whether they lie within the circle. Can you think of an implementation that does not use

arrays? How many points do you need to sample to consistently get results that agree with the true answer (π) to within 10%?

Put the commented code in a file called `monte_carlo1.py`.

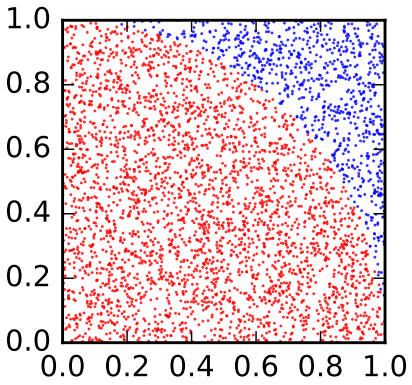

Figure 1: Random points on a unit square. Red points have a distance less than 1 from (0,0), whereas blue points are further away.

3 Mandelbrot set II

Rewrite your function `in_mandelbrot` from the last assignment to use a `for` loop. Make use of the `break` keyword to avoid computing additional elements of the series z_n once you know it will diverge ($|z_n| > 2$).

Put the commented code in a file called `mandelbrot2.py`.