Lab Project 1: How random is random?

Almost all good computer programs contain at least one random-number generator.
– D.E. Knuth (1969)

Write Python code to accomplish the following:

A. Create random number sequences:

Construct three different 1000 element random number sequences on the interval [0,1), using the random number generation schemes below.

i. Use Python numpy.random.uniform random number generator. Use a different seed for each sequence.

ii. Use John von Neumann's first suggested deterministic pseudo-random number technique – start with a six-digit number (the seed), square it and use the middle six digits of the result as your next entry, repeat. Start with three different seeds to create three random number sequences, normalize them each by 1.0e06. NOTE: this method has a tendency to fall into a trap of repeated numbers. *Try different seeds, plot the random number sequences and describe what visually non-random behaviors you get.*

iii. Write your code to check for these problems and restart the sequence with a different seed should they occur.

iv. EXTRA: How random are numbers "thought up at random?" Think up 1000 numbers between [0,63] and normalizing them by 64. If you chose to do this write the data to disk so that you can reload it for later use and do not have to generate the series each time. You may also want to do this once and exchange with two other people in the class rather than dream up a different sequence three times. *What is the significance of 64 here? What problems might be introduced in later analysis by using numbers between* [0,63]*?*

B. Test random number sequences:

i. Plot the random number series: Label axes, label the plots a) through f) [or i) if you did part A(iii)]. Three plots per page would make sense.
Are there features noticeable to the eye in the random number sequences?

ii. Make histograms of the runs separately and of all the values from all three of the series of each of generation technique together: Use separately 0.1 wide and 0.025 wide bins, label axes and plots. Four plots per page for each generation technique would make sense, with perhaps the different bin widths shown in different colors.

iii. Evaluate critically:

a. Frequency test:

i. Compare the observed and expected PDFs via a χ^2 measure of their difference for histograms using 10, 40, and 100 bins.

ii. *Tabulate* your results and *discuss* whether the method “passes” or “fails” the test. Base your assessment on the measured p-value. *Are there enough numbers in your sequence for this test to be valid (by rule of thumb)?*

b. Serial test:

i. Using 0.1 wide bins, count the number of times pairs of neighboring values in the sequence occur in the same bin. In neighboring bins?

ii. Make a χ^2 comparison with the expected values, testing each bin and neighboring pairs separately. *Discuss.*

iii. EXTRA: Extend this test beyond neighboring bin values to test how many times neighboring values in the sequence have any specified pair of bin values (k-distributed with k=2). Extend it further to test how many times any three neighboring values have any three bin values (k-distributed with k=3). *Do you have enough numbers in your sequence for this last test to be useful (by rule of thumb)? Hint: With 10 bin values, how many possible 3 bin sequences are there?*

Thought questions:

1. What does it mean for a sequence of numbers to be pseudorandom? What properties would you hope that it has? Do your random sequences exhibit any of these?
2. If you did not do Part A(iii), what biases might you expect to enter if you constructed the time series by thinking up numbers? If you did Part A(iii) did you think up the numbers or randomly hit keys on a keypad? Might there be significant differences? Might the biases be different? Did any of the biases show up in the quantitative evaluations?
3. Does it mean something different for a relatively short sequence to look random compared to a very long one? What problem could occur if you chose a short snippet of a very long truly random sequence as your set of random numbers?
4. Aside from engineering practicality, why might scientists choose a pseudo random number generator over a radioactive source coupled with a Geiger counter for their random number generator?

Comment on Project:

Please conclude your project report with one line (or more) of feedback. An "all good," or a comment if you find something that you think should be changed. This will help us tremendously in writing follow-on projects for this class and in revising the projects for future iterations of the class.