Due Friday October 23rd, 2020.

– – – – – – – – – – – – – – – – ⋆ – – – – – – – – – – – – – –

1. **python etc.**

   (a) *Bit operations on negative integers:* Negative integers in computer languages are usually bit-represented using a system called *two's complements*. Find out how this system works. Explain the results of the following expressions.

   (i) `(-5) >> 2`

   (ii) `~(-5)`

   (iii) `(-3) << 1`

   (iv) `(-2) & (-5)`

   (b) numpy arrays have a feature called **broadcasting** — figure out details. In the code

```
aa = np.array(range(12)).reshape(3,4)
bb = np.array(range(5,2,-1)).reshape(3,1)
aa + bb
aa*bb
```

   the arrays aa and bb are shaped as a $3 \times 4$ matrix and a $3 \times 1$ column vector. The last two expressions should supposedly be their elementwise sum and elementwise product. But they have different sizes!

   Explain why the last two expressions give no error and what the results should be.

2. **Distribution of combinations of random variables.**

   Consider independent random variates $X_0,\ X_1,\ \ldots$ each uniformly distributed on the support $[0, 1)$.

   (a) Derive the distribution of the sum $Y = X_0 + X_1$. This can be done using a *convolution*.

   (b) Check your derived distribution against a numerical calculation/histogram.

   (c) Derive the distribution of the product $Z = X_0 X_1$. This can also be done using convolutions.

   (d) Check your derived distribution for $Z = X_0 X_1$ against a numerical calculation/histogram. Is it convenient to plot one of the axes on a logarithmic scale?

   (e) Consider the sum of six uniform-distributed variates

   $$W = \sum_{i=0}^{5} X_i \ = \ X_0 + X_1 + X_2 + X_3 + X_4 + X_5$$

   How can this variable be transformed so that it gives an approximation to a **normally** distributed random variable? (Reminder: a normal distribution is a gaussian distribution with zero mean and unit standard deviation.)

   In what ways is this approximation poor? (Hint: generate a histogram and look at it.)

3. **Power method for eigenvalue and eigenvector**

   (a) You are given a $N \times N$ matrix $A$ and a random $(N \times 1)$ starting vector $x_0$. Find out and explain how the power method allows you to calculate a particular eigenvalue (and the corresponding eigenvector) of $A$. Which eigenvalue is determined?

   Note: the algorithm is listed in various ways; some descriptions might look trickier than others. But the basic idea is very simple, so if it looks complicated in one source, just try reading another.

   (b) Generate a $5 \times 5$ symmetric matrix with gaussian-random elements. By repeatedly multiplying a $5 \times 1$ vector, check (and show) if your estimate for the eigenvalue gradually converges.