extra problems

Ben Bolker

26 September 2019

## Matrix diagonal

1. Extract the diagonal of a square matrix
2. Extract the specified **off-diagonal** values of a square matrix.

e.g.

$$x=\left(\begin{matrix}x\_{11}&x\_{12}&x\_{13}&x\_{14}\\x\_{21}&x\_{22}&x\_{23}&x\_{24}\\x\_{31}&x\_{32}&x\_{33}&x\_{34}\\x\_{41}&x\_{42}&x\_{43}&x\_{44}\end{matrix}\right)$$

* mdiag(x) = [x11, x22, x33, x44]
* mdiag(x,-1) = [x12,x23,x34]
* mdiag(x,1) = [x21, x32, x43]

## Matrix multiplication

The formal definition of matrix multiplication for two matrices $x$ and $y$ is

$$(xy)\_{ij}=\sum\_{k=1}^{m}x\_{ik}y\_{kj}$$

where $m$ is the number of columns of $x$ and the number of rows of $y$ (these dimensions must be the same, or the matrices are *non-conformable* and can’t be multiplied).

As an example, if

$$x=\left(\begin{matrix}a&b\\c&d\end{matrix}\right),  y=\left(\begin{matrix}e&f\\g&h\end{matrix}\right)$$

then their product $z$ is

$$\left(\begin{matrix}ae+bg&af+bh\\ec+dg&cf+dh\end{matrix}\right)$$

## Remove-all

* Write a remove\_all(x,v) function that removes *all* occurrences of a value v within a list x and returns the result. It should still work (i.e. it should return x unchanged) if there are no occurrences in the list.

**example:**

x = ['a','z','y','z']
remove\_all(x,'z')
## answer: returns
['a','y']

## primes

1. Write a function prime1 that tests whether a specified natural number n is prime and returns a boolean value. In order to do this, you need to test whether it is divisible by any number between 2 and $\sqrt{n}$ (hint: your range should end at round(math.sqrt(n))+1; don’t forget to import math).
2. Write a function allprimes that uses a for or while loop and calls prime1 to generate and return a list of all of the primes between 2 and a specified natural number n (inclusive)

More efficient: [Sieve of Eratosthenes](https://www.geeksforgeeks.org/python-program-for-sieve-of-eratosthenes/)