CYRR 304 **Name:**
Homework 5, Spring 2024

"Is it hard?' Not if you have the right attitudes. It's having the right attitudes that's hard."
ROBERT PIRSIG

Homework 5 has questions 1 through 2 with a total of of 20 points. Your recorded score will be scaled to twenty points. The point value for each question or part of a question is in the box following each question or part of a question. This work is due **Saturday 2 March** at 11:59 PM.

For this assignment, convert your Jupyter notebook (a IPYNB file) to HTML and submit the HTML file to Canvas.

10 1. Write a Julia function `PR` (page rank) that does fixed point iteration on a stochastic matrix. This is a simplified version of the Google Page Rank algorithm. Specifically, for a stochastic matrix M, the function should find the terms of the recursively defined sequence p defined as

$$p_k = \begin{cases} p_0 & k = 0 \\ M p_{k-1} & k \in \mathbf{Z}_{\geq 1} \end{cases}.$$

For p_0, use a column vector whose elements are all given by $1/n$, where n is the number of rows of the matrix M. The Julia function `size` returns the size of a matrix as a two-tuple; for example

```
julia> size([1/2 1/3 0; 0 1/3 1/2; 1/2 1/3 1/2])
       (3,3)
```

To construct p_0, use a Julia array comprehension; for example

```
julia> mat = [1/2 1/3 0; 0 1/3 1/2; 1/2 1/3 1/2];

julia> (m,n) = size(mat);

julia> p0 = [[1/m] for k=1:m]

3-element Vector{Vector{Float64}}:
 [0.3333333333333333]
 [0.3333333333333333]
 [0.3333333333333333]
```

The function `PR` should stop finding terms of the fixed point sequence when consecutive terms p_k and p_{k+1} satisfy $\|p_k - p_{k+1}\|_\infty < \text{tol}$, where the positive number tol is a user-supplied tolerance. To find the infinity norm, use the function `norm`. To use this function, you will need to use the Julia package manager to install the package `LinearAlgebra`. After you have installed and loaded this package, you can find the infinity norm using

```
julia> norm([[1],[-4], [-2024]],Inf)
     2024
```

When the stopping criteria is never satisfied, your code should give an error "Fixed point sequence diverges or converges too slowly."

Let's give our function function keyword arguments to both the stopping tolerance `tol` and for the maximum number of iterations `maxiter`. Specifically, define your function as

```
function PR(mat::Matrix; tol = sqrt(eps(eltype(mat))), maxiter::Int64 = 50)
```

10 2. Test your function `PR` on the four inputs

```
PR([1/2 1/3 0; 0 1/3 1/2; 1/2 1/3 1/2])
3-element Vector{Vector{Float64}}:
 [0.22222222753502474]
 [0.3333333285777581]
 [0.4444444438872168]

 PR([0.0 1.0; 1.0 0.0])
 2-element Vector{Vector{Float64}}:
 [0.5]
 [0.5]

 PR( [1 0.1 0 0 ; 0 0 0 1; 0 0 1 0; 0 0.9 0 0])
 Fixed point sequence diverges or converges too slowly.

 PR( [1 0.1 0 0 ; 0 0 0 1; 0 0 1 0; 0 0.9 0 0], maxiter=1000)
 4-element Vector{Vector{Float64}}:
 [0.7499997441900677]
 [1.3463680648771707e-7]
 [0.25]
 [1.2117312583894537e-7]
```

Your values should be very close to the given values; if not, check your work to part 'a,' and try again.