```
julia> R10=rand(10,10)
10×10 Matrix{Float64}:
 0.195913    0.221536  0.123329  …  0.282942   0.455429
 0.21155     0.327569  0.980772     0.0722493  0.522395
 0.657984    0.84527   0.918299     0.0186169  0.512767
 0.923268    0.777922  0.832054     0.798341   0.78788
 0.00771684  0.846096  0.421965     0.218007   0.71076
 0.382188    0.577925  0.388981  …  0.493099   0.416712
 0.290013    0.100249  0.230561     0.83717    0.1258
 0.665663    0.778672  0.18233      0.534912   0.903102
 0.683753    0.844825  0.482785     0.32306    0.270413
 0.00795036  0.888788  0.53225      0.32845    0.840261
```

```
julia> x=ones(10);

julia> b=R10*x;

julia> xapp=R10\b
10-element Vector{Float64}:
 0.9999999999999971
 1.0000000000000138
 0.9999999999997571
 0.9999999999999848
 1.0000000000000446
 1.0000000000001659
 0.999999999999991
 1.0000000000001397
 0.9999999999999857
 0.9999999999998285
```

4 digits off

```
julia> using LinearAlgebra

julia> cond(R10)
7844.480384629356
```

condition number is a little less than 10^4

Question: how big on average is the condition number of a 10×10 matrix

Question: how big on average is the condition number of a 11×11 matrix

⋮

Question: how big on average is the condition number of a $n \times n$ matrix

Extra Credit: Graph κ_n vs. n and conjecture a functional dependency of κ_n on n.
Also how does it depend on which matrix norm is used?
Extra Extra Credit: Prove the relation.

Create a function

$$\kappa_n = \langle \mathrm{cond}(R_n) : R_n \text{ is a random } n \times n \text{ matrix} \rangle$$

average over maybe 10,000 matrices..

How to compute $\|A\|_2 = \max\{\|Ax\|_2 : \|x\|_2 = 1\}$

Start w/ Simple Idea...

$$\|Ax\|_2^2 = Ax \cdot Ax = A^T Ax \cdot x = Bx \cdot x$$

Spectral Theorem for B: Since $B = B^T$ then there is an orthonormal basis of eigenvectors of B.

$$B\xi_i = \lambda_i \xi_i \quad \text{and} \quad \xi_i \cdot \xi_j = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

Order the eigenvalues so $0 \leq \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$.

Given $x \in \mathbb{R}^n$ with $\|x\|_2 = 1$ plug it in

write x with respect to the basis ξ_i. $\boxed{x = \underbrace{c_1\xi_1 + c_2\xi_2 + \cdots + c_n\xi_n}_{x_0}}$

$$\|Ax\|_2^2 = Bx \cdot x = B(c_1\xi_1 + c_2\xi_2 + \cdots + c_n\xi_n) \cdot (c_1\xi_1 + c_2\xi_2 + \cdots + c_n\xi_n)$$

$$= (\boxed{\underbrace{c_1\lambda_1\xi_1 + c_2\lambda_2\xi_2 + \cdots + c_n\lambda_n\xi_n}_{y_1}}) \cdot (c_1\xi_1 + c_2\xi_2 + \cdots + c_n\xi_n)$$

$$= c_1^2\lambda_1 + c_2^2\lambda_2 + \cdots + c_n^2\lambda_n$$

Algorithm. iterate B

Let $x_0 \in \mathbb{R}^n$

$y_1 = Bx_0$

$x_1 = y_1/\|y_1\|$

$y_2 = Bx_1$

$y_{k+1} = Bx_k$

$x_{k+1} = y_{k+1}/\|y_{k+1}\|$

Claim: This algorithm will allow us to find $\|A\|_2$. How?

$$y_2 = B\left(\frac{c_1\lambda_1\xi_1 + c_2\lambda_2\xi_2 + \cdots + c_n\lambda_n\xi_n}{\|c_1\lambda_1\xi_1 + c_2\lambda_2\xi_2 + \cdots + c_n\lambda_n\xi_n\|}\right)$$

$$= \frac{B(c_1\lambda_1\xi_1 + c_2\lambda_2\xi_2 + \cdots + c_n\lambda_n\xi_n)}{\|c_1\lambda_1\xi_1 + c_2\lambda_2\xi_2 + \cdots + c_n\lambda_n\xi_n\|}$$

$$= \frac{c_1\lambda_1^2\xi_1 + c_2\lambda_2^2\xi_2 + \cdots + c_n\lambda_n^2\xi_n}{\|c_1\lambda_1\xi_1 + c_2\lambda_2\xi_2 + \cdots + c_n\lambda_n\xi_n\|}$$

$$x_2 = \frac{\dfrac{c_1\lambda_1^2\xi_1 + c_2\lambda_2^2\xi_2 + \cdots + c_n\lambda_n^2\xi_n}{\cancel{\|c_1\lambda_1\xi_1 + c_2\lambda_2\xi_2 + \cdots + c_n\lambda_n\xi_n\|}}}{\left\|\dfrac{c_1\lambda_1^2\xi_1 + c_2\lambda_2^2\xi_2 + \cdots + c_n\lambda_n^2\xi_n}{\cancel{\|c_1\lambda_1\xi_1 + c_2\lambda_2\xi_2 + \cdots + c_n\lambda_n\xi_n\|}}\right\|} = \frac{c_1\lambda_1^2\xi_1 + c_2\lambda_2^2\xi_2 + \cdots + c_n\lambda_n^2\xi_n}{\|c_1\lambda_1^2\xi_1 + c_2\lambda_2^2\xi_2 + \cdots + c_n\lambda_n^2\xi_n\|}$$

Therefore

$$x_k = \frac{c_1\lambda_1^k\xi_1 + c_2\lambda_2^k\xi_2 + \cdots + c_n\lambda_n^k\xi_n}{\|c_1\lambda_1^k\xi_1 + c_2\lambda_2^k\xi_2 + \cdots + c_n\lambda_n^k\xi_n\|}$$

Now take limits

λ_n is the largest

$$\lim_{k\to\infty} x_k = \lim_{k\to\infty} \frac{c_1\lambda_1^k\xi_1 + c_2\lambda_2^k\xi_2 + \cdots + c_n\lambda_n^k\xi_n}{\|c_1\lambda_1^k\xi_1 + c_2\lambda_2^k\xi_2 + \cdots + c_n\lambda_n^k\xi_n\|}$$

$$= \lim_{k\to\infty} \frac{c_1(\lambda_1/\lambda_n)^k\xi_1 + c_2(\lambda_2/\lambda_n)^k\xi_2 + \cdots + c_{n-1}\left(\frac{\lambda_{n-1}}{\lambda_n}\right)^k\xi_{n-1} + c_n\xi_n}{\|c_1(\lambda_1/\lambda_n)^k\xi_1 + c_2(\lambda_2/\lambda_n)^k\xi_2 + \cdots + c_{n-1}\left(\frac{\lambda_{n-1}}{\lambda_n}\right)^k\xi_{n-1} + c_n\xi_n\|}$$

it could happen that this ratio is also 1.

so in the limit the only terms that survive are when $\lambda_j/\lambda_n = 1$

$$= \frac{c_p\xi_p + c_{p+1}\xi_{p+1} + \cdots + c_n\xi_n}{\|c_p\xi_p + c_{p+1}\xi_{p+1} + \cdots + c_n\xi_n\|}$$

This is another eigenvector.

$$B\,\frac{c_p\lambda_n\xi_p + c_{p+1}\lambda_n\xi_{p+1} + \cdots + c_n\lambda_n\xi_n}{\|c_p\xi_p + c_{p+1}\xi_{p+1} + \cdots + c_n\xi_n\|} = \lambda_n \frac{c_p\xi_p + c_{p+1}\xi_{p+1} + \cdots + c_n\xi_n}{\|c_p\xi_p + c_{p+1}\xi_{p+1} + \cdots + c_n\xi_n\|}$$

Now solve for λ_n and we have $\|A\|_2 = \lambda_n^{1/2}$

How to solve for λ_n?

Idea $Bx = \lambda x$ and we know x but not λ.

$$\underbrace{Bx\cdot x}_{\text{scalar}} = \lambda \underbrace{x\cdot x}_{\text{scalar unit vector}} \qquad \lambda = \frac{Bx\cdot x}{x\cdot x} = Bx\cdot x$$

```
julia> B=R10'*R10
10×10 Matrix{Float64}:
 2.50942  2.74663  2.278
```

```
julia> xn=rand(10); xn=xn/norm(xn)
       for n=1:10
           yn=B*xn
           println("Bxn dot xn =",yn'*xn)
           xn=yn/norm(yn)
       end
Bxn dot xn =18.52256108072416
Bxn dot xn =24.008495914440637
Bxn dot xn =24.013025101802683
Bxn dot xn =24.013037116851983
Bxn dot xn =24.013037202622314
Bxn dot xn =24.013037203375166
Bxn dot xn =24.013037203381977
Bxn dot xn =24.013037203382044
Bxn dot xn =24.01303720338204
Bxn dot xn =24.013037203382037
```

```
julia> sqrt(24.013037203382037)
4.900309908912092

julia> norm(R10)
5.49744590533285

julia> opnorm(R10)
4.900309908912093
```

← wrong norm

← $\|R10\|_2$

These are the same.