

# Random Matrix Music Maker

Daniel Jordán

Remote.Python.Pizza 2020 🍕

# About me

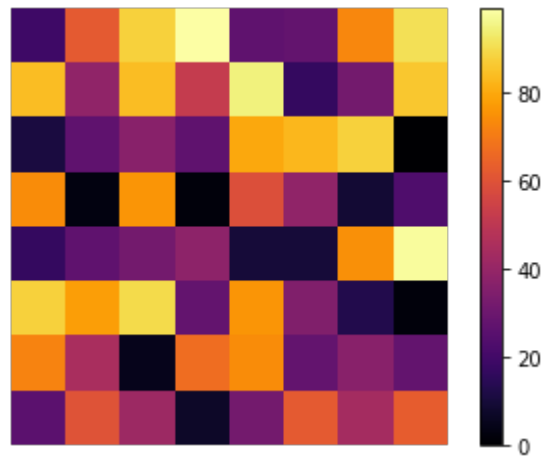
- I'm from Peru 🇵🇪
- Studied classical guitar 🎸 & physics ⚗️
- Musical influences: numpy . random module
- [github.com/danj](https://github.com/danj)

# Origin story (of this project)

- Needed lists of random numbers for master's thesis
- Started playing around with 2D arrays of random numbers

```
In [5]: A = np.random.randint(100, size=(8,8))  
printmatrix(A)
```

```
[19  62  88  99  27  28  73  91]  
[84  39  84  52  95  17  32  86]  
[11  27  37  27  80  83  88   0]  
[74   3  76   2  59  39   9  23]  
[17  27  32  38  10  10  75  98]  
[88  78  90  28  76  35  13   2]  
[72  45   5  67  74  28  37  28]  
[26  60  42   7  32  62  44  63]
```

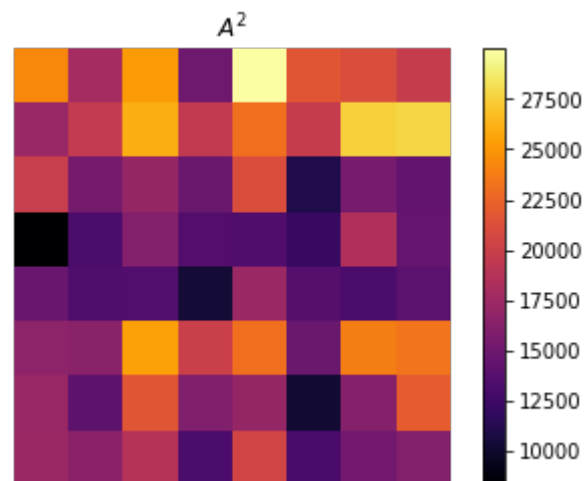


- Raising the matrix to the  $n$ -th power:

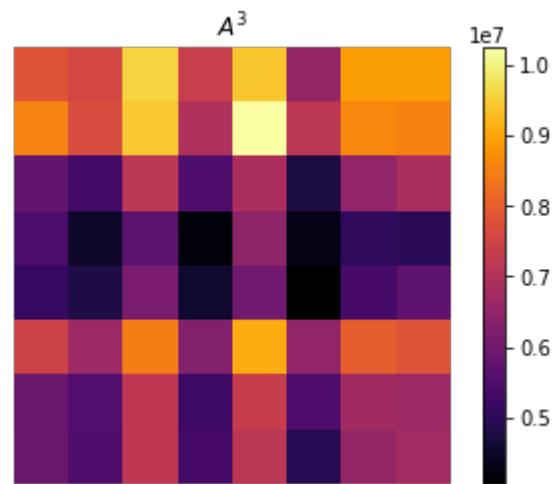
$$A^n = \underbrace{A \times A \times \cdots \times A}_{n\text{-times}}$$

```
In [7]: n = 2  
printmatrix(Apwr[n], title='$A^{'+str(n)+'}$')
```

24408	17927	25231	15017	29996	21687	21100	19817
17295	19644	26070	19548	23123	19788	27554	27852
19882	15409	17127	14806	21035	11060	15501	14414
8323	13183	16137	13636	13437	12142	18481	14599
14753	13390	13547	10398	17372	13699	13230	14017
16646	16499	25485	20007	23084	14825	23816	23368
17275	14082	21642	16008	17074	10301	16254	22071
17320	16567	18884	13183	20479	13135	15183	16148

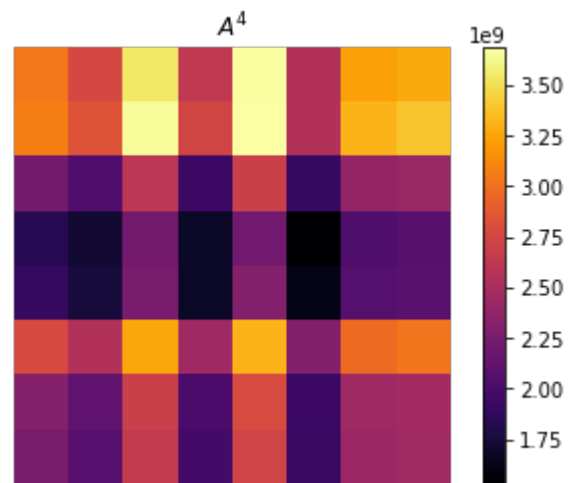


```
In [8]: n = 3  
printmatrix(Apwr[n], 4, 4, title='$A^{'+str(n)+'}$')
```

$$\begin{bmatrix} 7811249 & 7578735 & 9578127 & 7359370 & \dots \\ 8554498 & 7679775 & 9450704 & 6950499 & \\ 5777866 & 5333492 & 7154340 & 5510102 & \\ 5459211 & 4524230 & 5701528 & 4163466 & \\ \vdots & & & & \ddots \end{bmatrix}$$


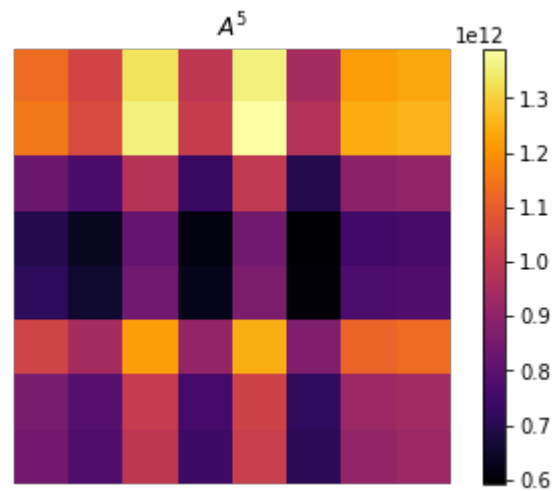
```
In [9]: n = 4  
printmatrix(Apwr[n], 4, 4, title='$A^{'+str(n)+'}$')
```

```
[ 3043692892  2761372486  3547575239  2640120458  ...  
 3071817741  2840886809  3650261488  2740857693  
 2227392233  2039435861  2610105786  1933419224  
 1837941779  1718955057  2217224181  1678498078  
      ⋮                      ⋱
```



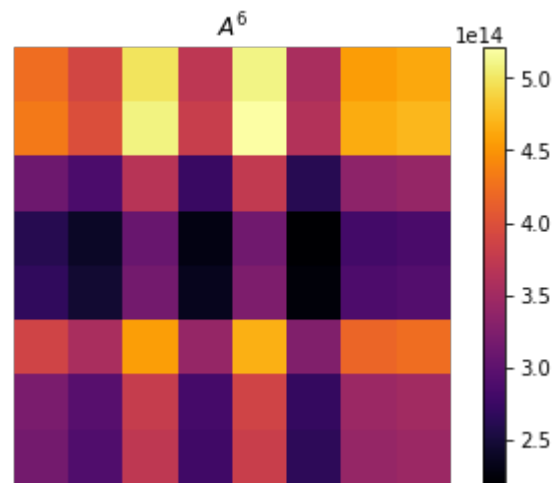


```
In [10]: n = 5  
printmatrix(Apwr[n], 3, 3, title='$A^{' + str(n) + '} $')
```

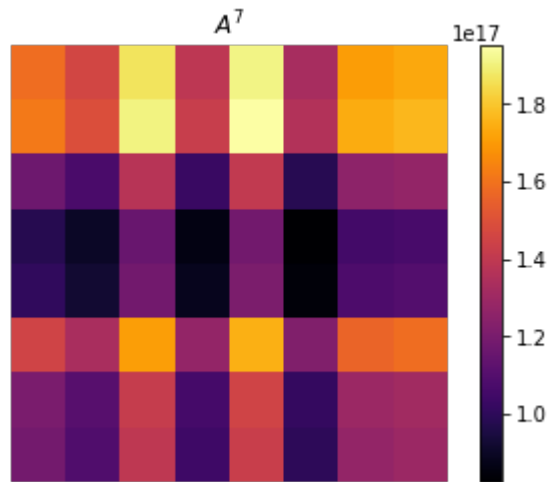
$$\begin{bmatrix} 1129558952849 & 1040744667692 & 1332937575010 & \dots \\ 1154213577855 & 1059424415610 & 1358827052368 & \\ 834218647739 & 767976011998 & 981956038755 & \\ \vdots & & & \ddots \end{bmatrix}$$


```
In [11]: n = 6  
printmatrix(Apwr[n], 3, 3, title='$A^{' + str(n) + '}$')
```

```
[ 423864298833651  389415838659178  498877097766995  ...  
 432425936944752  397632568004538  509140955267776  
 312283263793120  286941991462561  367771192564753  
      ⋮                                ⋱ ]
```



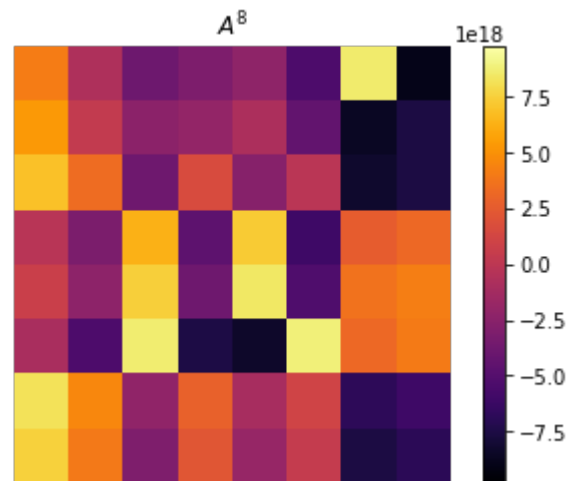
```
In [12]: n = 7  
printmatrix(Apwr[n], 2, 2, title='$A^{' + str(n) + '} $')
```

$$\begin{bmatrix} 158699889098875296 & 145890103767796344 & \dots \\ 161968285483597774 & 148867396115839362 & \\ \vdots & & \ddots \end{bmatrix}$$


★ We've reached a *steady state*!... but why?

- Except...

```
In [13]: n = 8
printmatrix(Apwr[n], 2, 2, title='$A^{'+str(n)+'}$')
```

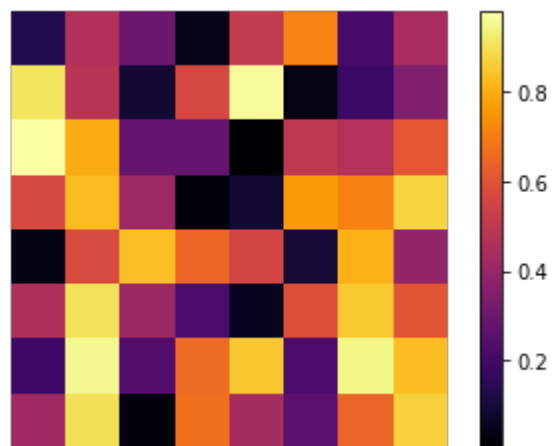
$$\begin{bmatrix} 4104101061765071831 & -701051311985645928 & \dots \\ 5320917576895901972 & 419338100412461312 & \\ \vdots & & \ddots \end{bmatrix}$$


OVERFLOW!

- Relative magnitudes are important

```
In [14]: A = np.random.random(size=(8,8))  
printmatrix(A)
```

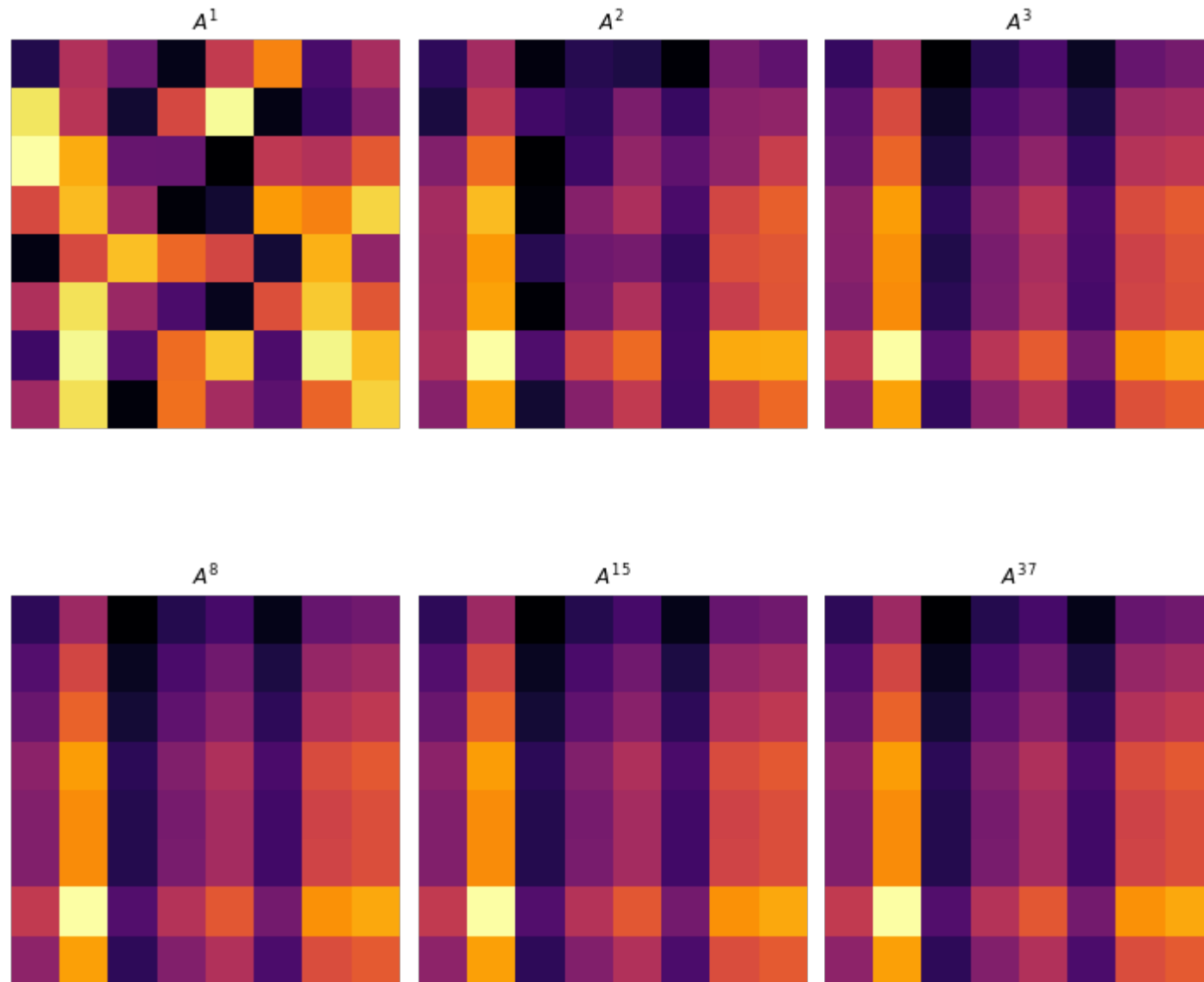
0.133	0.466	0.297	0.049	0.512	0.714	0.218	0.443
0.913	0.486	0.090	0.564	0.966	0.040	0.187	0.350
0.980	0.798	0.287	0.286	0.005	0.501	0.472	0.611
0.566	0.829	0.417	0.021	0.089	0.763	0.713	0.880
0.035	0.570	0.837	0.652	0.556	0.096	0.811	0.390
0.460	0.904	0.410	0.222	0.055	0.584	0.856	0.605
0.195	0.957	0.242	0.662	0.850	0.227	0.952	0.833
0.421	0.901	0.026	0.674	0.438	0.263	0.641	0.872



- Same matrix to various powers

In [16]: fig

Out[16]:





## And now, music?

- Pattern similar to a *step sequencer*
- Each square button is a tone

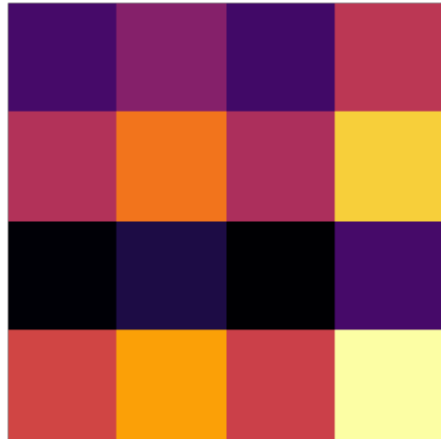




- Similar to a  $4 \times 4$  matrix

In [18]: `fig #4x4 matrix to the 10th power`

Out[18]:



- I followed tutorials by Alicia Key ([github.com/akey7](https://github.com/akey7))
  - Used sounddevice module
    - Created Random Matrix Music Maker( $RM^3$ ):
      - (Apologies to those with Perfect Pitch)

```
In [19]: from rm3 import rm3
         wave = rm3()
         wave.play()

         wave.save('wave1.wav')

         display(Audio('wave1.wav'))
```



0:00:00 / 13:31:36



- A tool for musical inspiration

In [20]: `wave.make_matrix()`

```
[[812.00040874 701.1679452 578.13058906 566.43540346]  
 [587.32008402 504.75173227 413.09101224 404.37808286]  
 [348.49000064 295.96579508 237.65778236 232.11509142]  
 [421.84275582 360.0910058 291.53937469 285.02293635]]
```



- You can make and play new matrices
- Lighter squares generate higher-frequency tones while darker squares generate lower-frequency tones

```
In [21]: wave.make_play(show=True)

wave.save('wave2.wav')

display(Audio('wave2.wav'))
```

```
[[260.27655152 326.23104733 542.95381432 284.56284276]
 [256.13348557 321.3032898  535.39819881 280.12876348]
 [427.96125036 526.27246501 849.26779673 464.16008129]
 [368.57657767 455.42888986 740.78834818 400.5565973  ]]
```



0:00:00 / 13:31:36



- We can shift frequencies to closest notes:

```
In [22]: wave.to_notes()  
  
         wave.save('wave2_tonotes.wav')  
  
         display(Audio('wave2_tonotes.wav'))
```

C4 E4 C#5 C#4 C4 E4 C5 C#4 A4 C5 G#5 A#4 F#4 A#4 F#5 G4



0:00:00 / 13:31:36



- Can also change various parameters
- For example, a *bass-y* riff

```
In [23]: wave.tempo = 70
         wave.central_freq = 100
         wave.freq_spread = 'tight'
         wave.clip_level = 0.2
         wave.make_play()

         wave.save('wave3.wav')

         display(Audio('wave3.wav'))
```



0:00:00 / 13:31:36



- Or a chaotic lead

```
In [24]: wave.tempo = 120  
         wave.central_freq = 500  
         wave.freq_spread = 'wide'  
         wave.clip_level = 0.2  
         wave.make_play()  
  
         wave.save('wave4.wav')  
  
         display(Audio('wave4.wav'))
```



0:00:00 / 13:31:36



- If you like it, repeat it or loop it!

```
In [25]: wave.make_play(2, show=True)

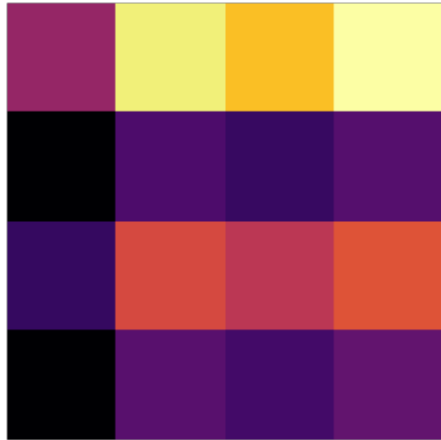
wave.save('wave5.wav', 2) # Saved melody 2 times

display(Audio('wave5.wav'))
```

```
[ [ 506.5685784  1167.67821135 1045.85927782 1220.1543463 ]
  [   20.          292.49942744  233.13781793  318.0709174 ]
  [ 226.29668589  710.24340593  621.06952175  748.65721344]
  [   20.          327.59637729  265.72994357  354.24672781]]
```



0:00:00 / 13:31:36





- Or use it to make some interesting effects

```
In [26]: wave.dimension = 7
wave.tempo = 800
wave.central_freq = 567
wave.freq_spread = 'wide'
wave.clip_level = 0.35
wave.make_play(2, show=True)
```

```
wave.save('wave6.wav', 2)
```

```
display(Audio('wave6.wav'))
```

```
[[ 713.63243065  591.41519231  336.72900565  607.85296713  488.07992095
   450.39914318   20.          ]
 [1461.12286598 1283.44954059  913.20409087 1307.34504639 1133.23032555
  1078.45123825  345.79387818]
 [ 562.77018775  451.74458832  220.38147194  466.68206176  357.87150864
   323.64639642   20.          ]
 [ 959.01615754  818.60168957  525.96115326  837.4932083   699.84830086
   656.56177028   77.52338839]
 [1089.87999444  939.75153686  626.89127049  959.95029193  812.80575722
   766.52636874  147.44288798]
 [1081.37833622  931.87767254  620.33709311  951.99604634  805.47203233
   759.39102798  142.90070566]
 [ 122.67115692   44.29458937   20.          54.83453995   20.
    20.          20.          ]]
```



0:00:00 / 13:31:36



# Thank you!

- Find project at [github.com/danj7/RandomMatrixMusicMaker/](https://github.com/danj7/RandomMatrixMusicMaker/)
- I'll be in the Discord! 🙌