In[•]:=
```
2^8
```

Out[•]=
```
256
```

In[•]:=
```
RandDataList1 = Table[RandomInteger[{1, 10}], {i, 1, 256}]
RandDataList2 = Table[RandomInteger[{1, 100}], {i, 1, 256}]
RandDataList3 = Table[RandomInteger[{1, 1000}], {i, 1, 256}]
```

Out[•]=
```
{9, 10, 3, 3, 5, 6, 6, 8, 4, 8, 10, 8, 1, 7, 9, 8, 10, 10, 10, 7, 6, 8, 3, 5, 7, 3, 5, 9,
 1, 3, 1, 5, 2, 4, 5, 9, 5, 5, 2, 7, 8, 9, 3, 7, 9, 8, 4, 7, 6, 6, 9, 5, 2, 2, 2, 8, 9, 1,
 9, 3, 8, 6, 10, 4, 3, 9, 6, 6, 3, 5, 1, 7, 5, 10, 1, 4, 5, 3, 9, 9, 7, 9, 9, 4, 1, 10, 1,
 8, 8, 6, 1, 10, 6, 6, 5, 9, 9, 10, 1, 10, 3, 6, 1, 8, 4, 9, 8, 9, 1, 10, 8, 8, 1, 4, 8,
 3, 3, 6, 6, 2, 2, 3, 1, 2, 4, 7, 1, 7, 7, 2, 2, 3, 3, 5, 5, 5, 1, 3, 9, 5, 4, 7, 3, 9, 9,
 10, 9, 1, 4, 10, 6, 4, 5, 7, 10, 7, 10, 10, 1, 4, 5, 5, 5, 7, 2, 2, 6, 3, 9, 1, 10, 3,
 1, 10, 10, 10, 4, 10, 10, 5, 9, 6, 2, 3, 4, 10, 7, 9, 2, 2, 3, 7, 1, 5, 7, 7, 7, 9, 5,
 1, 10, 7, 8, 5, 1, 5, 8, 7, 7, 9, 9, 8, 2, 6, 4, 9, 4, 8, 5, 5, 9, 8, 6, 4, 6, 6, 8, 4,
 2, 4, 5, 6, 7, 8, 1, 6, 10, 6, 4, 1, 2, 8, 10, 4, 3, 1, 4, 9, 5, 10, 5, 2, 4, 1, 1, 4}
```

Out[•]=
```
{80, 29, 79, 94, 59, 48, 21, 68, 87, 44, 50, 75, 3, 57, 52, 100, 8, 88, 63, 78, 55, 16, 52, 79, 49,
 52, 37, 58, 61, 23, 97, 66, 33, 15, 38, 74, 26, 95, 48, 54, 52, 68, 58, 58, 53, 3, 41, 86,
 15, 61, 6, 9, 64, 21, 51, 58, 22, 90, 90, 34, 69, 79, 80, 11, 33, 27, 92, 77, 91, 90, 44,
 66, 34, 13, 7, 85, 2, 38, 37, 27, 16, 81, 88, 39, 36, 53, 70, 20, 24, 46, 43, 2, 54, 85, 68,
 71, 34, 21, 28, 6, 44, 18, 68, 62, 62, 35, 93, 45, 39, 19, 68, 85, 8, 60, 54, 57, 49, 46,
 88, 60, 29, 80, 72, 62, 38, 95, 86, 89, 64, 63, 13, 24, 49, 21, 41, 92, 19, 27, 28, 97, 29,
 34, 93, 45, 56, 84, 61, 98, 92, 71, 48, 69, 43, 78, 39, 75, 93, 24, 22, 65, 35, 7, 45, 69,
 29, 98, 31, 64, 89, 21, 1, 28, 43, 76, 51, 48, 12, 10, 45, 16, 94, 18, 59, 50, 64, 21, 73,
 82, 68, 36, 97, 94, 92, 84, 60, 61, 52, 56, 95, 48, 31, 31, 2, 32, 11, 53, 27, 16, 93, 64,
 48, 20, 36, 48, 94, 72, 9, 97, 33, 64, 81, 71, 69, 46, 78, 60, 60, 62, 26, 46, 90, 70, 87,
 37, 45, 46, 32, 73, 12, 74, 36, 30, 16, 97, 66, 23, 72, 59, 74, 62, 31, 38, 65, 82, 72, 36}
```

Out[•]=
```
{516, 349, 232, 254, 133, 170, 41, 288, 750, 86, 144, 410, 810, 482, 762, 286, 349, 184, 593,
 650, 629, 5, 967, 853, 718, 646, 62, 918, 48, 45, 524, 438, 153, 864, 17, 37, 353, 354,
 524, 28, 937, 194, 335, 301, 547, 205, 454, 696, 727, 582, 937, 835, 127, 732, 892, 601,
 617, 969, 984, 325, 554, 605, 84, 464, 736, 250, 222, 978, 319, 13, 65, 570, 478, 446, 810,
 910, 717, 891, 197, 771, 869, 325, 529, 8, 816, 537, 761, 521, 842, 469, 276, 638, 110,
 927, 933, 92, 649, 767, 44, 538, 261, 686, 130, 981, 658, 419, 355, 390, 381, 581, 764,
 164, 454, 452, 855, 359, 926, 999, 167, 206, 586, 533, 561, 562, 55, 204, 112, 39, 442,
 143, 345, 616, 539, 450, 392, 700, 62, 677, 633, 574, 899, 920, 992, 627, 661, 709, 819,
 263, 86, 220, 956, 151, 12, 357, 226, 380, 385, 75, 409, 242, 950, 408, 271, 65, 133, 14,
 800, 961, 431, 617, 445, 923, 144, 459, 12, 195, 777, 381, 781, 720, 659, 138, 416, 880,
 652, 800, 759, 555, 898, 588, 302, 722, 753, 843, 123, 646, 960, 649, 396, 556, 467, 298,
 719, 614, 419, 915, 715, 341, 235, 54, 137, 459, 25, 110, 357, 591, 559, 372, 921, 137,
 683, 445, 553, 331, 243, 209, 591, 71, 941, 974, 55, 88, 352, 831, 369, 547, 337, 209,
 297, 605, 807, 756, 897, 377, 813, 800, 823, 566, 709, 278, 642, 328, 115, 827, 275, 754}
```

In[•]:=
```
RandDataList4 = Table[RandomInteger[{1, 10000}], {i, 1, 256}]
RandDataList5 = Table[RandomInteger[{1, 100000}], {i, 1, 256}]
```

Out[•]=
```
{6492, 9304, 2616, 4097, 3429, 3400, 7776, 9486, 2693, 1960, 3791, 5694, 7774, 3789, 4115,
 9494, 169, 4185, 8248, 8505, 6841, 2909, 5675, 7558, 9287, 9202, 6540, 205, 1304, 2216,
 4596, 8445, 1889, 2699, 6918, 7746, 8449, 5434, 5761, 1051, 4805, 9367, 1250, 7615,
 4177, 2062, 8685, 2038, 5182, 7939, 7799, 8771, 1244, 4229, 9096, 565, 3168, 5518,
 2871, 8499, 6778, 2707, 977, 7765, 8080, 7201, 8331, 7116, 3052, 8996, 3649, 6553,
 147, 4581, 7896, 5427, 124, 9288, 9419, 1956, 3691, 2771, 8631, 9655, 3163, 9423,
 1183, 61, 9106, 1055, 7772, 2190, 6440, 2290, 5072, 9963, 8336, 7082, 1025, 2577,
 5714, 4948, 4757, 9864, 2088, 4373, 8476, 236, 4204, 3171, 7486, 3754, 338, 4884,
 1115, 1455, 6328, 2129, 9081, 7045, 2762, 7319, 4129, 4463, 8860, 140, 5069, 5468,
 5751, 9486, 4237, 7588, 2692, 8513, 9859, 1970, 984, 2688, 1743, 4227, 2171, 131,
 2177, 5222, 7054, 3263, 793, 5323, 1144, 8431, 4071, 4063, 8183, 6671, 8927, 3052,
 5185, 5432, 6793, 3675, 3474, 9936, 6970, 9034, 6587, 8353, 1587, 9944, 624, 398,
 9679, 8596, 2798, 3309, 2429, 8274, 1164, 5670, 8126, 3851, 1849, 1752, 4314, 8594,
 8401, 4271, 4177, 8919, 5178, 2204, 1328, 5829, 2692, 1937, 5002, 35, 2408, 5900, 984,
 8187, 5077, 7085, 425, 2648, 1337, 1037, 2130, 7041, 4211, 9057, 8949, 9446, 9962,
 6290, 8329, 53, 3007, 6496, 3451, 2352, 916, 7681, 7543, 4851, 4424, 7137, 6308, 674,
 1714, 7097, 2488, 5844, 4888, 533, 5630, 7348, 1760, 5837, 4550, 6825, 2015, 1905,
 1255, 9651, 9718, 6556, 9703, 8855, 5847, 8665, 2427, 5804, 526, 6086, 5844, 4772}
```

Out[•]=
```
{5952, 30991, 32130, 43751, 90502, 14745, 36926, 1278, 85957, 27377, 99179, 30861,
 88279, 85870, 19528, 90419, 95006, 8741, 34664, 20069, 93779, 82592, 42001, 48976,
 57884, 96416, 79073, 53845, 8558, 55681, 25177, 65261, 24347, 34105, 75714, 13536,
 42032, 73996, 46395, 59959, 22187, 1138, 83151, 29252, 9038, 4152, 70552, 22507,
 765, 32370, 36300, 35510, 63137, 16203, 94190, 96306, 97365, 27919, 50199, 90496,
 51034, 82597, 50447, 70030, 26339, 1062, 24435, 1595, 4749, 71846, 13482, 88078,
 81841, 24376, 83536, 89157, 68478, 87564, 27338, 96411, 62754, 7779, 32930, 83589,
 66472, 80745, 63492, 7807, 35800, 80230, 85425, 20726, 21091, 33518, 32758, 73327,
 58707, 20832, 88782, 24636, 41608, 86035, 56596, 32763, 13867, 56319, 91483,
 56339, 97264, 64278, 50498, 15485, 62167, 7850, 19611, 96890, 94304, 9327, 94116,
 72889, 44550, 56793, 12707, 15523, 40737, 64043, 93666, 74780, 23881, 61372,
 18332, 45809, 74644, 22981, 39055, 19282, 55510, 42632, 48952, 57570, 71194,
 71352, 98417, 43504, 24754, 13324, 84074, 20647, 10590, 36734, 88530, 56305,
 82615, 81474, 89707, 14375, 91898, 13952, 17837, 21698, 6776, 66378, 74169, 65573,
 68073, 39710, 27229, 53773, 11259, 16751, 10176, 27470, 7942, 53139, 57625, 11518,
 75259, 75345, 31326, 87138, 32904, 59780, 82720, 80192, 13722, 45844, 67144,
 69447, 64640, 82350, 42241, 89248, 25682, 45297, 5614, 37239, 19281, 25568, 63388,
 29619, 55350, 81491, 89830, 45892, 71949, 981, 8126, 63589, 48894, 85177, 10416,
 57504, 6955, 47135, 47616, 50237, 68889, 82321, 92046, 52185, 81590, 1863, 57002,
 23038, 42133, 56399, 89597, 14725, 68252, 24053, 11375, 18015, 44347, 54719,
 68187, 80439, 3893, 12210, 85087, 42809, 54585, 28713, 82774, 60747, 91174,
 75010, 30110, 82391, 75058, 60604, 2833, 63922, 27421, 45317, 57965, 11758}
```

In[•]:=
```
Length[RandDataList1]
```

Out[•]= 256

```
In[]:= RandDataList0 = Table[RandomInteger[{1, 1}], {i, 1, 256}]

Out[]= {1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1}
```

Pre

Run these

```
SVDRandDataVec =
{{8, 7, 1, 7, 6, 9, 3, 8, 1, 9, 9, 8, 10, 5, 2, 2, 3, 10,
   5, 1, 5, 10, 1, 5, 9, 10, 4, 2, 9, 4, 2, 2, 8, 4, 6, 4, 3, 4, 10, 8, 8,
   9, 3, 10, 5, 2, 10, 8, 8, 7, 3, 9, 8, 1, 2, 1, 10, 8, 6, 9, 1, 3, 7, 9},
  {4, 75, 74, 90, 34, 3, 52, 83, 87, 70, 23, 34, 82, 41, 41, 58, 94, 16, 18, 12, 92,
   94, 34, 63, 90, 32, 66, 75, 3, 4, 7, 60, 70, 77, 66, 32, 99, 58, 89, 11, 80, 60, 22,
   88, 78, 8, 11, 6, 78, 18, 65, 26, 80, 95, 87, 30, 14, 48, 8, 21, 10, 20, 81, 58},
  {1000, 773, 179, 70, 120, 424, 809, 627, 553, 271, 946, 384, 525, 557, 572, 219,
   585, 942, 572, 847, 930, 567, 996, 534, 707, 542, 855, 108, 389, 715, 377, 9,
   529, 848, 976, 390, 333, 863, 108, 329, 339, 825, 828, 859, 145, 645, 734, 722,
   817, 665, 219, 13, 617, 874, 511, 230, 886, 659, 234, 507, 824, 48, 429, 17},
  {4910, 3417, 1269, 3176, 101, 3946, 1599, 5810, 4080, 5587, 5731, 8765,
   6933, 6913, 6884, 6313, 7732, 231, 9783, 1881, 664, 2813, 7393, 2747, 7379,
   740, 3703, 1484, 1577, 3411, 7708, 8604, 1175, 191, 3505, 3283, 7989, 9886,
   9702, 6857, 3533, 190, 9961, 4554, 7703, 2872, 9857, 3534, 6628, 3651, 2328,
   5611, 962, 908, 9030, 2756, 3559, 5908, 6911, 7927, 3829, 9826, 6209, 1637},
  {28 286, 34 422, 44 572, 77 603, 1487, 10 884, 50 424, 69 225, 21 821, 89 705, 7831,
   48 866, 67 177, 56 714, 51 509, 81 721, 64 105, 53 175, 72 465, 99 840, 70 700, 37 411,
   65 255, 62 977, 89 572, 12 078, 70 585, 79 063, 73 538, 12 638, 42 918, 88 844, 34 192,
   72 972, 11 738, 97 154, 44 239, 99 800, 67 726, 46 996, 7043, 1464, 70 893, 72 565,
   75 983, 86 170, 79 924, 38 877, 25 043, 98 651, 57 373, 20 546, 59 157, 25 354,
   98 714, 74 455, 71 108, 93 345, 46 089, 70 414, 49 399, 52 797, 68 082, 62 221}}
```

```
k = 2;

(* M= DataVec *)
```

In[ ]:=

```
(* k=1; *)

k = 5;
RandDataVec[[k]] = Table[RandomInteger[{1, 10^k}], {i, 1, 64}];

M = RandDataVec[[k]];

numrowsW = √Length[M] ;

W = Table[
    Table[M[[i]], {i, (((j - 1) * (numrowsW)) + 1), (j * (numrowsW))}], {j, 1, numrowsW}];
Print["Dimensions[W] = ", Dimensions[W]]

ρ = (W.Transpose[W]);
Print["Dimensions[ρ] = ", Dimensions[ρ]]

MatA = ρ ;

DataEigens = Diagonal[SingularValueDecomposition[MatA][[2]]] // N;
(* Sort[DeleteCases[Eigenvalues[ρ]//N,0.],Greater]; *)
(*DeleteCases Removes 0's from the set of Eigenvalues,
Sort puts the list in order of greatest to least *)

DataEigensSet = DataEigens / Total[DataEigens];
(* This is the set of nonzero normalized eigenvalues in order of greatest to least *)
set = Sort[DeleteCases[DataEigensSet // N, 0.], Greater];

Print["Total number of eigenvalues = ", Length[DataEigensSet]]
Print["Normalized Eigenvalues = ", DataEigensSet]

Print["Normalized nonzero Eigenvalues = ", set]

n = Length[set];
H[α_] := 1/(1 - α) Log[Sum[(set[[i]])^α, {i, 1, n}]] // N
H0 = Log[n] // N ;(* H0 = Hartley Entropy*)
H1 = -Sum[((set[[i]]) (Log[set[[i]]])), {i, 1, n}] // N ;(* H1 = Shannon Entropy*)
H2onward = Table[H[a], {a, 2, 20}] // N ;(* H2 onward *)

RenyiEntropyofEigenvalues = Join[{H0}, {H1}, H2onward];

Renyis = Table[{i - 1, RenyiEntropyofEigenvalues[[i]]}, {i, 1, 21}];

Print["Renyi Entropies Indexed = ", Renyis]

SVDDataEigensSetVector[[k]] = set;
SVDRenyisVector[[k]] = Renyis;
```

```
Dimensions[W] = {8, 8}

Dimensions[ρ] = {8, 8}

Total number of eigenvalues = 8

Normalized Eigenvalues =
 {0.849263, 0.0577148, 0.0443503, 0.02576, 0.0132935, 0.00776141, 0.00162046, 0.000236403}

Normalized nonzero Eigenvalues =
 {0.849263, 0.0577148, 0.0443503, 0.02576, 0.0132935, 0.00776141, 0.00162046, 0.000236403}

Renyi Entropies Indexed = {{0, 2.07944}, {1, 0.643337}, {2, 0.318211}, {3, 0.244835},
   {4, 0.217838}, {5, 0.204232}, {6, 0.196063}, {7, 0.190617}, {8, 0.186727}, {9, 0.183809},
   {10, 0.18154}, {11, 0.179725}, {12, 0.178239}, {13, 0.177002}, {14, 0.175954}, {15, 0.175057},
   {16, 0.174279}, {17, 0.173598}, {18, 0.172997}, {19, 0.172463}, {20, 0.171985}}
```

In[•]:=
```
Table[Length[SVDDataEigensSetVector[[i]]], {i, 1, 5}]
```

Out[•]=
```
{8, 8, 8, 8, 8}
```

In[•]:=
```
RandDataVec
SVDDataEigensSetVector
SVDRenyisVector
```

Out[•]=
```
{{8, 7, 1, 7, 6, 9, 3, 8, 1, 9, 9, 8, 10, 5, 2, 2, 3, 10, 5, 1, 5, 10, 1, 5, 9, 10, 4, 2, 9, 4, 2, 2,
   8, 4, 6, 4, 3, 4, 10, 8, 8, 9, 3, 10, 5, 2, 10, 8, 8, 7, 3, 9, 8, 1, 2, 1, 10, 8, 6, 9, 1, 3, 7, 9},
  {4, 75, 74, 90, 34, 3, 52, 83, 87, 70, 23, 34, 82, 41, 41, 58, 94, 16, 18, 12, 92,
   94, 34, 63, 90, 32, 66, 75, 3, 4, 7, 60, 70, 77, 66, 32, 99, 58, 89, 11, 80, 60, 22,
   88, 78, 8, 11, 6, 78, 18, 65, 26, 80, 95, 87, 30, 14, 48, 8, 21, 10, 20, 81, 58},
  {1000, 773, 179, 70, 120, 424, 809, 627, 553, 271, 946, 384, 525, 557, 572, 219,
   585, 942, 572, 847, 930, 567, 996, 534, 707, 542, 855, 108, 389, 715, 377, 9,
   529, 848, 976, 390, 333, 863, 108, 329, 339, 825, 828, 859, 145, 645, 734, 722,
   817, 665, 219, 13, 617, 874, 511, 230, 886, 659, 234, 507, 824, 48, 429, 17},
  {4910, 3417, 1269, 3176, 101, 3946, 1599, 5810, 4080, 5587, 5731, 8765, 6933,
   6913, 6884, 6313, 7732, 231, 9783, 1881, 664, 2813, 7393, 2747, 7379, 740,
   3703, 1484, 1577, 3411, 7708, 8604, 1175, 191, 3505, 3283, 7989, 9886, 9702,
   6857, 3533, 190, 9961, 4554, 7703, 2872, 9857, 3534, 6628, 3651, 2328, 5611,
   962, 908, 9030, 2756, 3559, 5908, 6911, 7927, 3829, 9826, 6209, 1637},
  {28 286, 34 422, 44 572, 77 603, 1487, 10 884, 50 424, 69 225, 21 821, 89 705, 7831,
   48 866, 67 177, 56 714, 51 509, 81 721, 64 105, 53 175, 72 465, 99 840, 70 700,
   37 411, 65 255, 62 977, 89 572, 12 078, 70 585, 79 063, 73 538, 12 638, 42 918, 88 844,
   34 192, 72 972, 11 738, 97 154, 44 239, 99 800, 67 726, 46 996, 7043, 1464, 70 893,
   72 565, 75 983, 86 170, 79 924, 38 877, 25 043, 98 651, 57 373, 20 546, 59 157, 25 354,
   98 714, 74 455, 71 108, 93 345, 46 089, 70 414, 49 399, 52 797, 68 082, 62 221}}
```

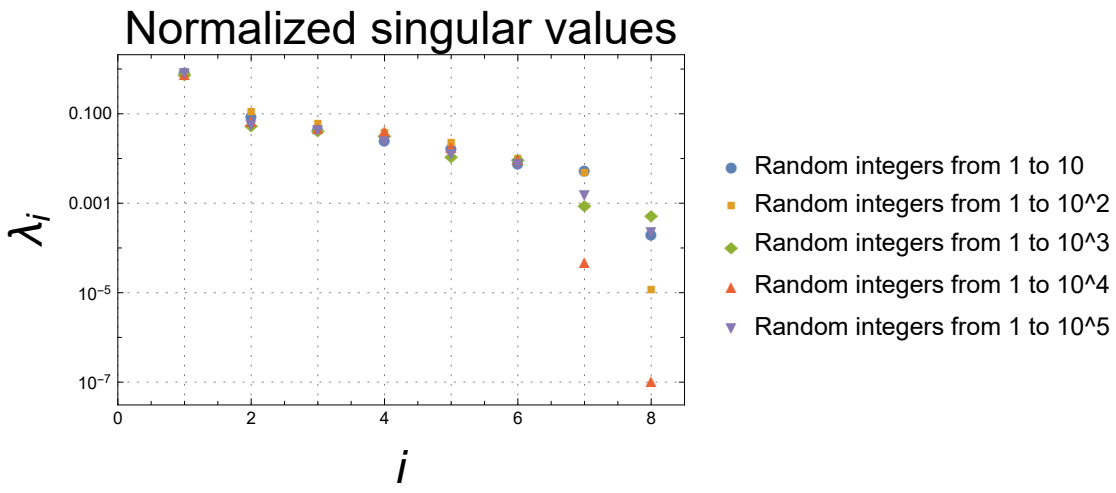
Out[•]=
```
{{0.817197, 0.0844948, 0.0439151, 0.0251836, 0.0162592,
   0.00750819, 0.00524087, 0.000200867}, {0.766272, 0.104587, 0.0581237,
   0.0352209, 0.0216947, 0.00960728, 0.00448284, 0.0000110242},
  {0.835105, 0.0587737, 0.0464849, 0.0356799, 0.0121033, 0.0103179,
   0.000973243, 0.000561915}, {0.809024, 0.0698419, 0.050291,
   0.0423015, 0.0185402, 0.00995171, 0.0000497091, 1.07225 × 10^-7},
  {0.849263, 0.0577148, 0.0443503, 0.02576, 0.0132935, 0.00776141, 0.00162046, 0.000236403}}
```

Out[•]= {{{0, 2.07944}, {1, 0.736668}, {2, 0.388812}, {3, 0.302163}, {4, 0.269125}, {5, 0.25234},
{6, 0.242249}, {7, 0.23552}, {8, 0.230714}, {9, 0.227109}, {10, 0.224305},
{11, 0.222062}, {12, 0.220227}, {13, 0.218697}, {14, 0.217403}, {15, 0.216294},
{16, 0.215333}, {17, 0.214492}, {18, 0.21375}, {19, 0.21309}, {20, 0.2125}},
{{0, 2.07944}, {1, 0.875451}, {2, 0.505318}, {3, 0.397778}, {4, 0.354828},
{5, 0.332759}, {6, 0.31946}, {7, 0.310587}, {8, 0.304249}, {9, 0.299495}, {10, 0.295797},
{11, 0.292839}, {12, 0.290419}, {13, 0.288402}, {14, 0.286696}, {15, 0.285233},
{16, 0.283965}, {17, 0.282856}, {18, 0.281877}, {19, 0.281007}, {20, 0.280229}},
{{0, 2.07944}, {1, 0.690199}, {2, 0.350206}, {3, 0.269995}, {4, 0.240251},
{5, 0.225246}, {6, 0.216237}, {7, 0.210231}, {8, 0.20594}, {9, 0.202722}, {10, 0.20022},
{11, 0.198217}, {12, 0.196579}, {13, 0.195214}, {14, 0.194059}, {15, 0.193069},
{16, 0.192211}, {17, 0.19146}, {18, 0.190798}, {19, 0.190209}, {20, 0.189682}},
{{0, 2.07944}, {1, 0.76181}, {2, 0.409234}, {3, 0.31737}, {4, 0.282543}, {5, 0.264907},
{6, 0.254312}, {7, 0.247248}, {8, 0.242202}, {9, 0.238418}, {10, 0.235474},
{11, 0.23312}, {12, 0.231193}, {13, 0.229587}, {14, 0.228229}, {15, 0.227065},
{16, 0.226055}, {17, 0.225172}, {18, 0.224393}, {19, 0.223701}, {20, 0.223081}},
{{0, 2.07944}, {1, 0.643337}, {2, 0.318211}, {3, 0.244835}, {4, 0.217838},
{5, 0.204232}, {6, 0.196063}, {7, 0.190617}, {8, 0.186727}, {9, 0.183809}, {10, 0.18154},
{11, 0.179725}, {12, 0.178239}, {13, 0.177002}, {14, 0.175954}, {15, 0.175057},
{16, 0.174279}, {17, 0.173598}, {18, 0.172997}, {19, 0.172463}, {20, 0.171985}}}

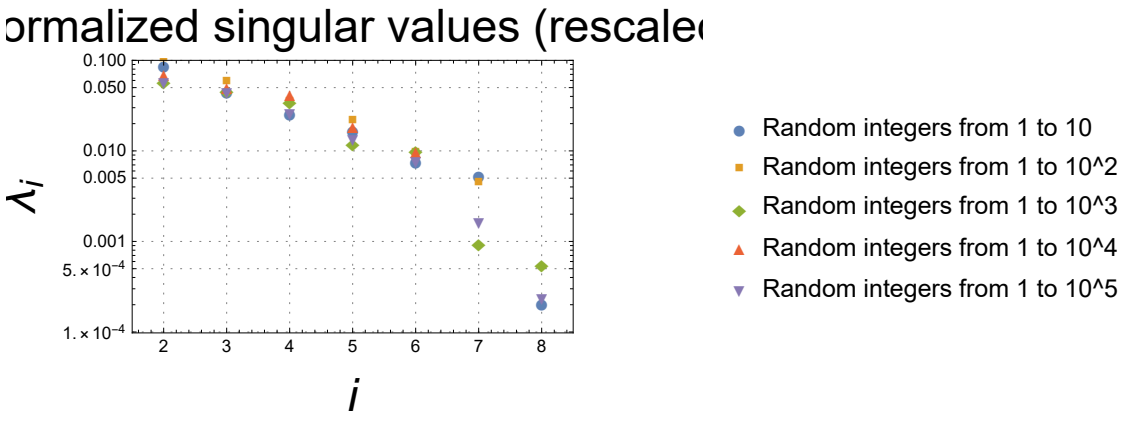
In[•]:=
```
ListLogPlot[SVDDataEigensSetVector, PlotRange → {{0, 8.5}, Automatic},
 PlotLabel → Style["Normalized singular values", 24],
 FrameLabel → {Style[i, 24], Style[λi, 24]},
 GridLines → {Table[i, {i, 0, Length[DataEigensSet]}], Automatic},
 GridLinesStyle → Directive[Gray, Dotted], Frame → True,
 PlotLegends → {"Random integers from 1 to 10", "Random integers from 1 to 10^2",
   "Random integers from 1 to 10^3", "Random integers from 1 to 10^4",
   "Random integers from 1 to 10^5"}, PlotMarkers → Automatic]
ListLogPlot[SVDDataEigensSetVector, PlotRange → {{1.5, 8.5}, {0, 0.1}},
 PlotLabel → Style["Normalized singular values (rescaled)", 24],
 FrameLabel → {Style[i, 24], Style[λi, 24]},
 GridLines → {Table[i, {i, 0, Length[DataEigensSet]}], Automatic},
 GridLinesStyle → Directive[Gray, Dotted], Frame → True,
 PlotLegends → {"Random integers from 1 to 10", "Random integers from 1 to 10^2",
   "Random integers from 1 to 10^3", "Random integers from 1 to 10^4",
   "Random integers from 1 to 10^5"}, PlotMarkers → Automatic]

ListPlot[SVDRenyisVector, PlotRange → {{-0.8, (Length[Renyis] - 0.5)}, All},
 PlotLabel → Style["Rényi Entropies", 24], FrameLabel → {Style[α, 24], Style[Sα, 24]},
 GridLines → {Table[i, {i, 0, Length[Renyis]}], Automatic},
 GridLinesStyle → Directive[Gray, Dotted],
 PlotLegends → {"Random integers from 1 to 10", "Random integers from 1 to 10^2",
   "Random integers from 1 to 10^3", "Random integers from 1 to 10^4",
   "Random integers from 1 to 10^5"}, Frame → True, PlotMarkers → Automatic]
ListPlot[SVDRenyisVector, PlotRange → {{-0.8, (Length[Renyis] - 0.5)}, {0, 1.0}},
 PlotLabel → Style["Rényi Entropies (rescaled)", 24],
 FrameLabel → {Style[α, 24], Style[Sα, 24]},
 GridLines → {Table[i, {i, 0, Length[Renyis]}], Automatic},
 GridLinesStyle → Directive[Gray, Dotted],
 PlotLegends → {"Random integers from 1 to 10", "Random integers from 1 to 10^2",
   "Random integers from 1 to 10^3", "Random integers from 1 to 10^4",
   "Random integers from 1 to 10^5"}, Frame → True, PlotMarkers → Automatic]
```

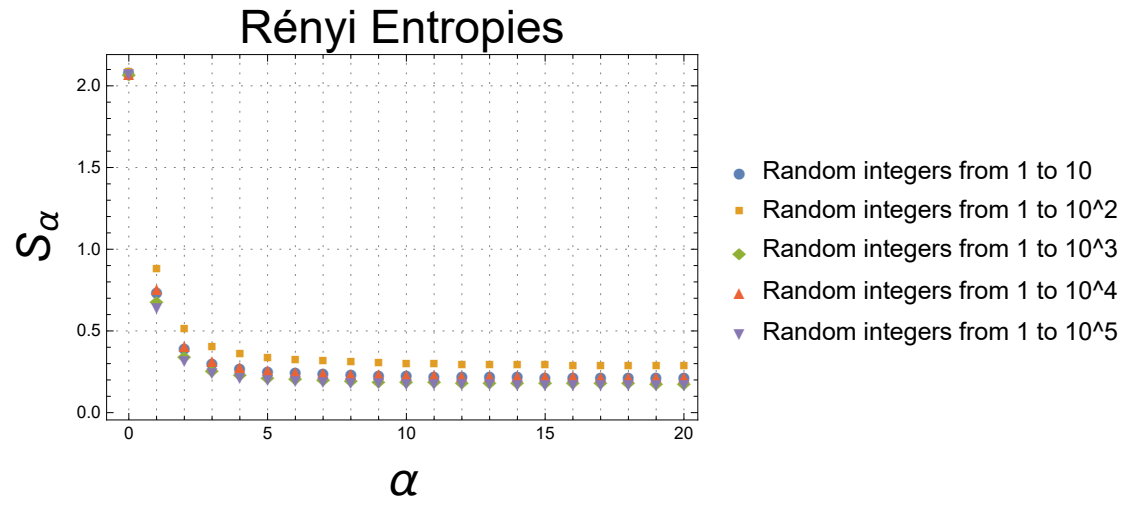
Out[•]=



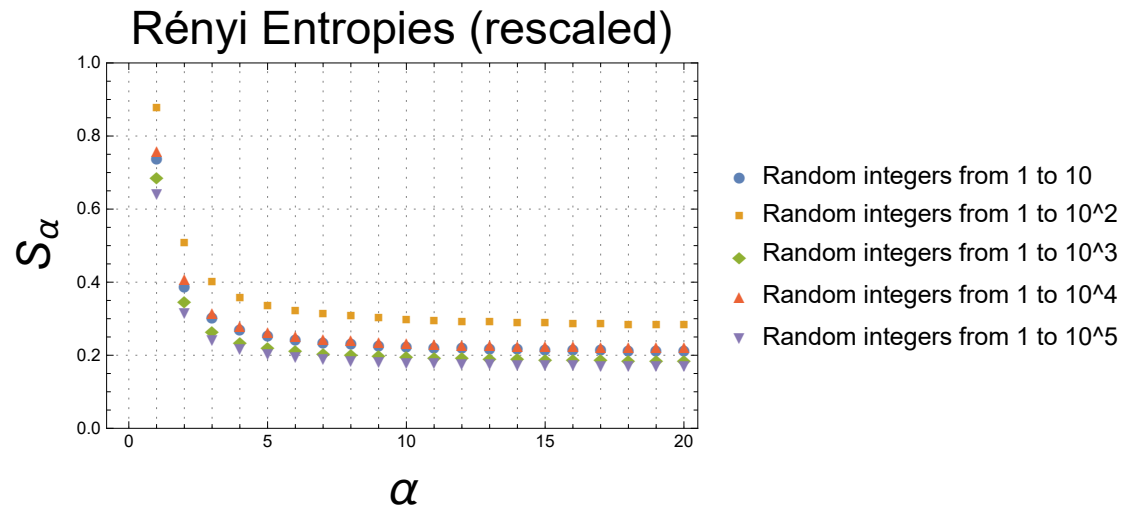
Out[ ]=



Out[ ]=



Out[ ]=



In[ ]:=
```
eDataEigensSetVector = {1, 1, 1, 1, 1};
eRenyisVector = {1, 1, 1, 1, 1};
```

In[ ]:=
```
(* k=1; *)
```

```
k = 5;


M = SVDRandDataVec[[k]];

(*DataVecName=StringJoin["RandDataList",ToString[k]];

Print ["Name of input list = ", DataVecName ]
 (* Print ["Name of input list = ", DataVecName ]*)

 M = ToExpression[DataVecName]; *)

numrowsW = √Length[M] ;

W = Table[
    Table[M[[i]], {i, (((j - 1) * (numrowsW)) + 1), (j * (numrowsW))}], {j, 1, numrowsW}];
Print["Dimensions[W] = ", Dimensions[W]]

ρ = (W.Transpose[W]);
Print["Dimensions[ρ] = ", Dimensions[ρ]]

MatA = ρ ;

DataEigens = Eigenvalues[MatA] // N;
(* Sort[DeleteCases[Eigenvalues[ρ]//N,0.],Greater]; *)
(*DeleteCases Removes 0's from the set of Eigenvalues,
Sort puts the list in order of greatest to least *)

DataEigensSet = DataEigens / Total[DataEigens];
(* This is the set of nonzero normalized eigenvalues in order of greatest to least *)
set = Sort[DeleteCases[DataEigensSet // N, 0.], Greater];

Print["Total number of eigenvalues = ", Length[DataEigensSet]]
Print["Normalized Eigenvalues = ", DataEigensSet]

Print["Normalized nonzero Eigenvalues = ", set]

n = Length[set];
H[α_] := 1/(1 - α) Log[Sum[(set[[i]])^α, {i, 1, n}]] // N
H0 = Log[n] // N ;(* H0 = Hartley Entropy*)
H1 = -Sum[((set[[i]])) ((Log[set[[i]]])), {i, 1, n}] // N ;(* H1 = Shannon Entropy*)
H2onward = Table[H[a], {a, 2, 20}] // N ;(* H2 onward *)

RenyiEntropyofEigenvalues = Join[{H0}, {H1}, H2onward];

Renyis = Table[{i - 1, RenyiEntropyofEigenvalues[[i]]}, {i, 1, 21}];

Print["Renyi Entropies Indexed = ", Renyis]
```

```
eDataEigensSetVector[[k]] = DataEigensSet;
eRenyisVector[[k]] = Renyis;
```

```
Dimensions[W] = {8, 8}
```

```
Dimensions[ρ] = {8, 8}
```

```
Total number of eigenvalues = 8
```

```
Normalized Eigenvalues =
 {0.849263, 0.0577148, 0.0443503, 0.02576, 0.0132935, 0.00776141, 0.00162046, 0.000236403}
```

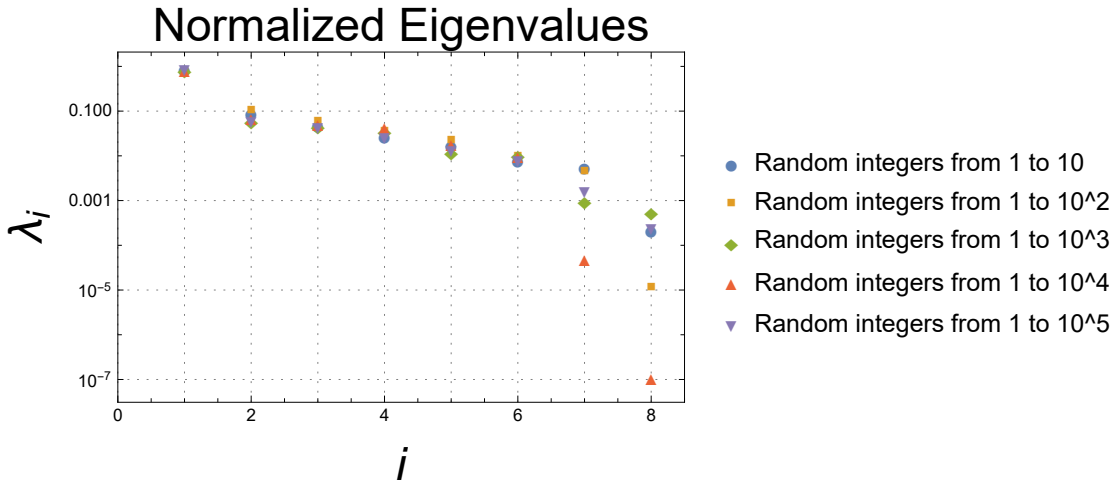
```
Normalized nonzero Eigenvalues =
 {0.849263, 0.0577148, 0.0443503, 0.02576, 0.0132935, 0.00776141, 0.00162046, 0.000236403}
```

```
Renyi Entropies Indexed = {{0, 2.07944}, {1, 0.643337}, {2, 0.318211}, {3, 0.244835},
  {4, 0.217838}, {5, 0.204232}, {6, 0.196063}, {7, 0.190617}, {8, 0.186727}, {9, 0.183809},
  {10, 0.18154}, {11, 0.179725}, {12, 0.178239}, {13, 0.177002}, {14, 0.175954}, {15, 0.175057},
  {16, 0.174279}, {17, 0.173598}, {18, 0.172997}, {19, 0.172463}, {20, 0.171985}}
```

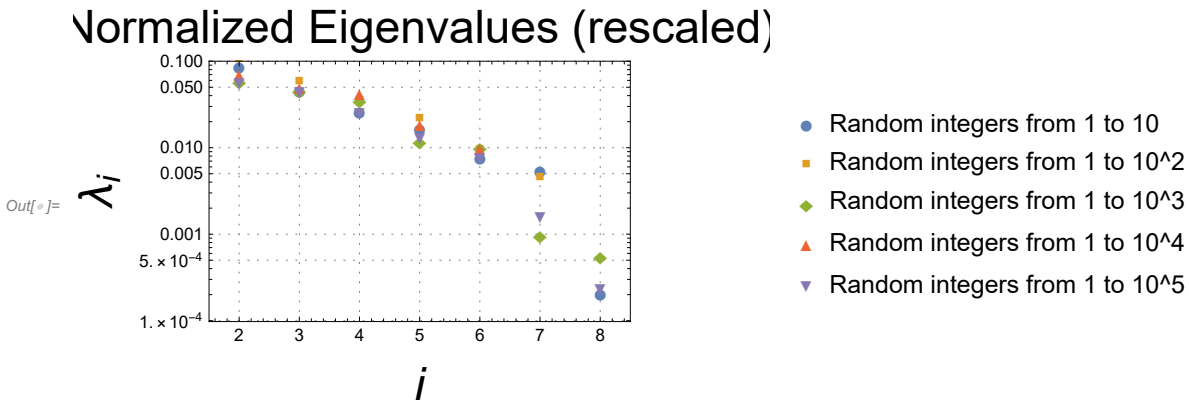
In[•]:=
```
ListLogPlot[eDataEigensSetVector,
 PlotRange → {{0, (Length[DataEigensSet] + 0.5)}, Automatic},
 PlotLabel → Style["Normalized Eigenvalues", 24],
 FrameLabel → {Style[i, 24], Style[λi, 24]},
 GridLines → {Table[i, {i, 0, Length[DataEigensSet]}], Automatic},
 GridLinesStyle → Directive[Gray, Dotted], Frame → True,
 PlotLegends → {"Random integers from 1 to 10", "Random integers from 1 to 10^2",
   "Random integers from 1 to 10^3", "Random integers from 1 to 10^4",
   "Random integers from 1 to 10^5"}, PlotMarkers → Automatic]
ListLogPlot[eDataEigensSetVector, PlotRange →
  {{1.5, (Length[DataEigensSet] + 0.5)}, {0, 0.1}},
 PlotLabel → Style["Normalized Eigenvalues (rescaled)", 24],
 FrameLabel → {Style[i, 24], Style[λi, 24]},
 GridLines → {Table[i, {i, 0, Length[DataEigensSet]}], Automatic},
 GridLinesStyle → Directive[Gray, Dotted], Frame → True,
 PlotLegends → {"Random integers from 1 to 10", "Random integers from 1 to 10^2",
   "Random integers from 1 to 10^3", "Random integers from 1 to 10^4",
   "Random integers from 1 to 10^5"}, PlotMarkers → Automatic]

ListPlot[eRenyisVector, PlotRange → {{-0.8, (Length[Renyis] - 0.5)}, All},
 PlotLabel → Style["Rényi Entropies", 24], FrameLabel → {Style[α, 24], Style[Sα, 24]},
 GridLines → {Table[i, {i, 0, Length[Renyis]}], Automatic},
 GridLinesStyle → Directive[Gray, Dotted],
 PlotLegends → {"Random integers from 1 to 10", "Random integers from 1 to 10^2",
   "Random integers from 1 to 10^3", "Random integers from 1 to 10^4",
   "Random integers from 1 to 10^5"}, Frame → True, PlotMarkers → Automatic]
ListPlot[eRenyisVector, PlotRange → {{-0.8, (Length[Renyis] - 0.5)}, {0, 1.0}},
 PlotLabel → Style["Rényi Entropies (rescaled)", 24],
 FrameLabel → {Style[α, 24], Style[Sα, 24]},
 GridLines → {Table[i, {i, 0, Length[Renyis]}], Automatic},
 GridLinesStyle → Directive[Gray, Dotted],
 PlotLegends → {"Random integers from 1 to 10", "Random integers from 1 to 10^2",
   "Random integers from 1 to 10^3", "Random integers from 1 to 10^4",
   "Random integers from 1 to 10^5"}, Frame → True, PlotMarkers → Automatic]
```

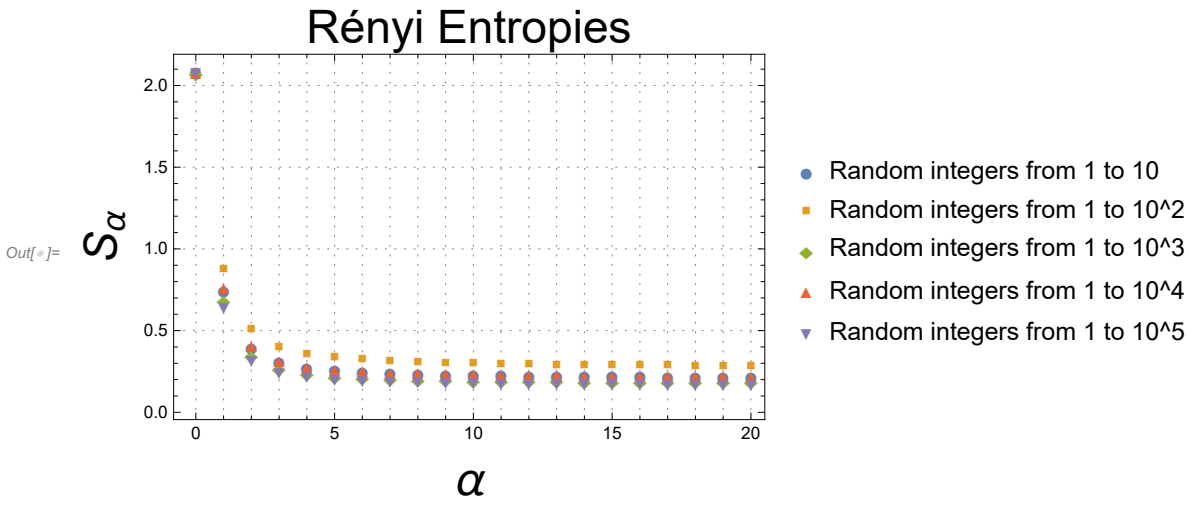
Out[•]=



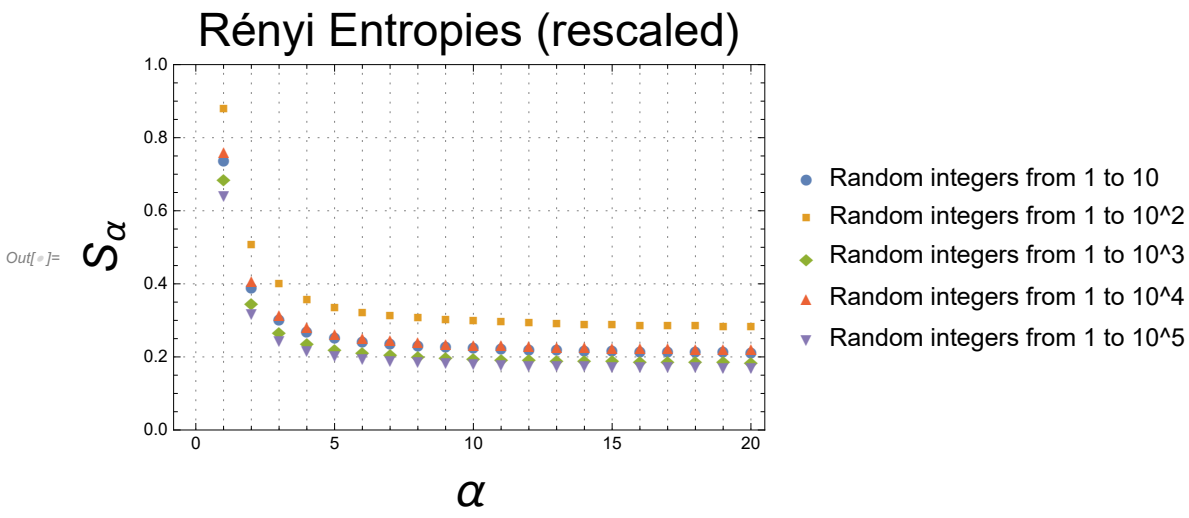
Out[ ]=



Out[ ]=



Out[ ]=



Compare eigendecomp to SVD for density matrix approach

Used same random vecs to get the decompositions with differences below:

In[•]:=
```
eDataEigensSetVector = SVDDataEigensSetVector
eRenyisVector = SVDRenyisVector
```

Out[•]=
$$\{\{0.,\,0.,\,0.,\,0.,\,0.,\,0.,\,0.,\,0.\},\ \{0.,\,0.,\,0.,\,0.,\,0.,\,0.,\,0.,\,0.\},\ \{2.22045\times10^{-16},\,0.,\,6.93889\times10^{-18},\,-6.93889\times10^{-18},\,0.,\,0.,\,0.,\,0.\},\ \{0.,\,0.,\,-6.93889\times10^{-18},\,0.,\,0.,\,0.,\,0.,\,0.\},\ \{0.,\,0.,\,0.,\,0.,\,0.,\,0.,\,0.,\,-2.71051\times10^{-20}\}\}$$

Out[•]=
$$\{\{\{0,0.\},\{0,0.\}\},$$
$$\{\{0,0.\},\{0,0.\}\},$$
$$\{\{0,0.\},\{0,-2.22045\times10^{-16}\},\{0,-4.44089\times10^{-16}\},\{0,-3.33067\times10^{-16}\},\{0,-3.05311\times10^{-16}\},\{0,-2.498\times10^{-16}\},\{0,-2.498\times10^{-16}\},\{0,-3.05311\times10^{-16}\},\{0,-2.498\times10^{-16}\},\{0,-2.498\times10^{-16}\},\{0,-2.498\times10^{-16}\},\{0,-2.498\times10^{-16}\},\{0,-2.77556\times10^{-16}\},\{0,-2.498\times10^{-16}\},\{0,-2.498\times10^{-16}\},\{0,-2.498\times10^{-16}\},\{0,-2.22045\times10^{-16}\},\{0,-2.22045\times10^{-16}\},\{0,-2.498\times10^{-16}\},\{0,-2.498\times10^{-16}\},\{0,-2.498\times10^{-16}\}\},$$
$$\{\{0,0.\},\{0,-1.11022\times10^{-16}\},\{0,0.\}\},$$
$$\{\{0,0.\},\{0,0.\}\}\}$$

In[•]:=
```
Eigenvalues[{{1, 1}, {0, 1}}] // N
Diagonal[SingularValueDecomposition[{{1, 1}, {0, 1}}][[2]]] // N
```

Out[•]= {1., 1.}

Out[•]= {1.61803, 0.618034}

In[•]:=
```
A = {{1, 2, 3}, {2, 1, 2}, {3, 2, 1}};
Eigenvalues[A] // N
Eigenvectors[A] // N
Diagonal[SingularValueDecomposition[A][[2]]] // N
```

Out[•]= {5.70156, -2., -0.701562}

Out[•]= {{1., 0.850781, 1.}, {-1., 0., 1.}, {1., -2.35078, 1.}}

Out[•]= {5.70156, 2., 0.701562}

In[•]:=
```
A // MatrixForm
```

Out[•]//MatrixForm=
$$\begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 2 \\ 3 & 2 & 1 \end{pmatrix}$$