

# Python Markov Transition Conditional Probability Check Sum to 1

Fan Wang

2020-12-03

# Contents

<b>1</b>	<b>Close Value Comparison</b>	<b>1</b>
1.1	Generate a Markov Transition Matrix . . . . .	1
1.2	Check Row by Row Tolerance . . . . .	4
1.3	Check Joint Tolerance . . . . .	5
1.4	Normalize Transition Matrix Row Sum . . . . .	6

## 1 Close Value Comparison

Go to the [RMD](#), [PDF](#), or [HTML](#) version of this file. Go back to [Python Code Examples Repository](#) ([bookdown site](#)) or the [pyfan](#) Package ([API](#)).

```
import sys as sys
import numpy as np
import pprint
```

### 1.1 Generate a Markov Transition Matrix

With numerical probability approximations, discrete random variables often are not exact, given an AR(1) transition matrix, how to check whether the conditional probabilities conditional on each current state (row) sums up to one?

First define the AR(1) transition matrix, which is from a model of asset transitions. This is a 50 by 50 matrix.

[illegible]

0.0005, 0.0001, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,  
0.,  
0., 0., 0.], [0., 0., 0., 0., 0.1408, 0.5552, 0.2403, 0.052, 0.0095, 0.0017,  
0.0003, 0.0001, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,  
0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,  
0., 0.], [0., 0., 0., 0., 0.1666, 0.5902, 0.2019, 0.0349, 0.0053, 0.0008,  
0.0001, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,  
0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,  
0.], [0., 0., 0., 0., 0., 0.2065, 0.5668, 0.1869, 0.0335, 0.0053, 0.0009, 0.0001,  
0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,  
0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,  
0.], [0., 0., 0., 0., 0.2414, 0.5453, 0.1748, 0.0321, 0.0053, 0.0009, 0.0002, 0., 0.,  
0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,  
0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.], [0., 0., 0., 0., 0.,  
0.0172, 0.4156, 0.4019, 0.1281, 0.0293, 0.0062, 0.0013, 0.0003, 0.0001, 0., 0.,  
0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,  
0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.], [0., 0., 0., 0., 0., 0.,  
0.0033, 0.3222, 0.4594, 0.1655, 0.0391, 0.0083, 0.0017, 0.0004, 0.0001, 0., 0.,  
0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,  
0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.], [0., 0., 0., 0., 0., 0., 0.,  
0.0111, 0.3744, 0.4215, 0.1471, 0.0357, 0.0079, 0.0017, 0.0004, 0.0001, 0., 0.,  
0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,  
0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.], [0., 0., 0., 0., 0., 0., 0., 0.,  
0.1555, 0.504, 0.2526, 0.0682, 0.0154, 0.0034, 0.0008, 0.0002, 0., 0., 0., 0.,  
0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,  
0., 0., 0., 0., 0., 0., 0., 0., 0., 0.], [0., 0., 0., 0., 0., 0., 0., 0., 0., 0.1987,  
0.5249, 0.215, 0.0493, 0.0097, 0.0019, 0.0004, 0.0001, 0., 0., 0., 0., 0.,  
0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,  
0., 0., 0., 0., 0., 0., 0., 0.], [0., 0., 0., 0., 0., 0., 0., 0., 0., 0.0041,  
0.3513, 0.4533, 0.1499, 0.0331, 0.0066, 0.0013, 0.0003, 0.0001, 0., 0., 0., 0.,  
0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,  
0., 0., 0., 0., 0., 0., 0., 0.], [0., 0., 0., 0., 0., 0., 0., 0., 0., 0.0247,  
0.441, 0.3835, 0.1174, 0.0264, 0.0055, 0.0012, 0.0003, 0.0001, 0., 0., 0.,  
0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,  
0., 0., 0., 0., 0., 0., 0., 0.], [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,  
0.1049, 0.502, 0.2887, 0.0809, 0.0183, 0.004, 0.0009, 0.0002, 0.0001, 0., 0.,  
0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,  
0., 0., 0., 0., 0., 0., 0., 0.], [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,  
0.0003, 0.1962, 0.4888, 0.2319, 0.0636, 0.0148, 0.0034, 0.0008, 0.0002, 0., 0.,  
0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,  
0., 0., 0., 0., 0., 0., 0., 0.], [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,  
0., 0.0087, 0.3362, 0.4294, 0.1674, 0.0444, 0.0105, 0.0025, 0.0006, 0.0001, 0.,  
0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,  
0., 0., 0., 0., 0., 0., 0., 0.], [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,  
0., 0.0001, 0.154, 0.4801, 0.2617, 0.0785, 0.0195, 0.0046, 0.0011, 0.0003,  
0.0001, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,  
0., 0., 0., 0., 0., 0., 0., 0.], [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,  
0., 0., 0.0001, 0.1467, 0.48, 0.2664, 0.0804, 0.02, 0.0048, 0.0012, 0.0003,  
0.0001, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,  
0., 0., 0., 0., 0., 0., 0., 0.], [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,  
0., 0., 0., 0.0149, 0.3543, 0.4103, 0.1614, 0.0444, 0.011, 0.0027, 0.0007,  
0.0002, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,  
0., 0., 0., 0., 0., 0., 0., 0.], [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,

0., 0., 0., 0., 0.0438, 0.4122, 0.3611, 0.1334, 0.0369, 0.0094, 0.0024, 0.0006,  
 0.0002, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,  
 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.], [0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,  
 0., 0., 0., 0., 0.1159, 0.452, 0.2939, 0.1007, 0.0278, 0.0072, 0.0019,  
 0.0005, 0.0001, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,  
 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.], [0., 0., 0., 0., 0., 0., 0., 0., 0.,  
 0., 0., 0., 0., 0., 0.0012, 0.198, 0.4463, 0.2436, 0.0804, 0.0224, 0.0059,  
 0.0016, 0.0004, 0.0001, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,  
 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.], [0., 0., 0., 0., 0., 0., 0., 0.,  
 0., 0., 0., 0., 0., 0., 0.0087, 0.2821, 0.4181, 0.2011, 0.0649, 0.0183, 0.005  
 , 0.0014, 0.0004, 0.0001, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,  
 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.], [0., 0., 0., 0., 0., 0., 0., 0.,  
 0., 0., 0., 0., 0., 0., 0.0842, 0.4174, 0.3217, 0.1244, 0.0376,  
 0.0105, 0.0029, 0.0008, 0.0002, 0.0001, 0., 0., 0., 0., 0., 0., 0., 0.,  
 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.], [0., 0., 0., 0., 0., 0., 0.,  
 0., 0., 0., 0., 0., 0., 0., 0., 0.0828, 0.4166, 0.3229, 0.1251, 0.0379,  
 0.0106, 0.0029, 0.0008, 0.0002, 0.0001, 0., 0., 0., 0., 0., 0., 0., 0.,  
 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.], [0., 0., 0., 0., 0., 0., 0.,  
 0., 0., 0., 0., 0., 0., 0., 0., 0.0119, 0.2907, 0.4075, 0.1979, 0.0656,  
 0.019, 0.0053, 0.0015, 0.0004, 0.0001, 0., 0., 0., 0., 0., 0., 0., 0.,  
 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.], [0., 0., 0., 0., 0., 0., 0.,  
 0., 0., 0., 0., 0., 0., 0., 0., 0.0424, 0.3654, 0.3618, 0.1581,  
 0.0513, 0.015, 0.0043, 0.0012, 0.0004, 0.0001, 0., 0., 0., 0., 0., 0., 0.,  
 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.], [0., 0., 0., 0., 0., 0., 0.,  
 0., 0., 0., 0., 0., 0., 0., 0., 0.0001, 0.1078, 0.4112, 0.3043,  
 0.1217, 0.0388, 0.0114, 0.0033, 0.001, 0.0003, 0.0001, 0., 0., 0., 0., 0.,  
 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.], [0., 0., 0., 0., 0., 0., 0.,  
 0., 0., 0., 0., 0., 0., 0., 0., 0.0015, 0.1714, 0.4155,  
 0.264, 0.1015, 0.0323, 0.0097, 0.0029, 0.0009, 0.0003, 0.0001, 0., 0., 0.,  
 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.], [0., 0., 0., 0., 0., 0., 0.,  
 0., 0., 0., 0., 0., 0., 0., 0., 0.0093,  
 0.2496, 0.4001, 0.2214, 0.082, 0.0261, 0.0079, 0.0024, 0.0007, 0.0002, 0.0001,  
 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.], [0., 0., 0., 0., 0., 0., 0.,  
 0., 0., 0., 0., 0., 0., 0., 0., 0.041, 0.3374, 0.3591, 0.1729, 0.0615, 0.0195, 0.006, 0.0018, 0.0006, 0.0002,  
 0.0001, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,  
 0.], [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,  
 0., 0.0295, 0.315, 0.3712, 0.186, 0.0673, 0.0215, 0.0066, 0.002, 0.0006,  
 0.0002, 0.0001, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,  
 0., 0., 0.], [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,  
 0., 0., 0., 0.0295, 0.315, 0.3712, 0.186, 0.0673, 0.0215, 0.0066, 0.002,  
 0.0006, 0.0002, 0.0001, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,  
 0., 0., 0., 0.], [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,  
 0., 0., 0., 0., 0.0036, 0.1927, 0.4021, 0.2527, 0.1004, 0.0334, 0.0104,  
 0.0032, 0.001, 0.0003, 0.0001, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,  
 0., 0., 0., 0., 0., 0.], [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,  
 0., 0., 0., 0., 0., 0., 0., 0.018, 0.2749, 0.3813, 0.2082, 0.0792, 0.0262,  
 0.0083, 0.0026, 0.0008, 0.0003, 0.0001, 0., 0., 0., 0., 0., 0., 0., 0.,  
 0., 0., 0., 0., 0., 0.], [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,  
 0., 0., 0., 0., 0., 0., 0., 0., 0.0539, 0.3414, 0.344, 0.1684, 0.0621,  
 0.0205, 0.0065, 0.0021, 0.0007, 0.0002, 0.0001, 0., 0., 0., 0., 0., 0.,  
 0., 0., 0., 0., 0., 0.], [0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,  
 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.0005, 0.1155, 0.3788, 0.2986,



one row that does not match this, then set condition for overall matrix checking to false.

```
fl_atol = 1e-08
print(f'with {fl_atol=}, many rows do not approximately equal to 1')

## with fl_atol=1e-08, many rows do not approximately equal to 1
print(f'[{np.allclose(fl_row_sum, fl_sum_to_match, rtol=0, atol=fl_atol) for fl_row_sum in ar_row_sums_ar1}]')

## [np.allclose(fl_row_sum, fl_sum_to_match, rtol=0, atol=fl_atol) for fl_row_sum in ar_row_sums_ar1]=[False]
print(f'{np.allclose(ar_row_sums_ar1, fl_sum_to_match, rtol=0, atol=fl_atol)=}')

## np.allclose(ar_row_sums_ar1, fl_sum_to_match, rtol=0, atol=fl_atol)=False
fl_atol = 1e-03
print(f'With {fl_atol=}, all rows approximately equal to 1')

## With fl_atol=0.001, all rows approximately equal to 1
print(f'[{np.allclose(fl_row_sum, fl_sum_to_match, rtol=0, atol=fl_atol) for fl_row_sum in ar_row_sums_ar1}]')

## [np.allclose(fl_row_sum, fl_sum_to_match, rtol=0, atol=fl_atol) for fl_row_sum in ar_row_sums_ar1]=[True]
print(f'{np.allclose(ar_row_sums_ar1, fl_sum_to_match, rtol=0, atol=fl_atol)=}')

## np.allclose(ar_row_sums_ar1, fl_sum_to_match, rtol=0, atol=fl_atol)=True
```

### 1.3 Check Joint Tolerance

For practical usages, we set joint condition. The difference between the sum of any row and 1 should be less than a threshold, additionally, the total average gap between row sums and 1 should be less than a threshold. Do this so that can have a more relaxed per-row tolerance requirement, and more stringent average requirement. Setting only a stringent per row requirement might be too restive, and reject transition matrixes that do not have problems.

```
# criteria
fl_atol_per_row = 1e-03
fl_atol_avg_row = 1e-04
# per-row check
bl_per_row_pass = np.allclose(ar_row_sums_ar1, fl_sum_to_match, rtol=0, atol=fl_atol_per_row)
# all-rows check
fl_row_gap_mean = np.mean(abs(ar_row_sums_ar1-fl_sum_to_match))
bl_avg_row_pass = np.allclose(fl_row_gap_mean+fl_sum_to_match, fl_sum_to_match, rtol=0, atol=fl_atol_avg_row)
# Joint
bl_ar1_sum_pass = bl_per_row_pass and bl_avg_row_pass
# Print
print(f'{bl_per_row_pass=}')

## bl_per_row_pass=True
print(f'{bl_avg_row_pass=} and {fl_row_gap_mean=}')

## bl_avg_row_pass=True and fl_row_gap_mean=6.8000000000001916e-05
print(f'{bl_ar1_sum_pass=}')

## bl_ar1_sum_pass=True
```

## 1.4 Normalize Transition Matrix Row Sum

Having pass the checks, would like to have conditional probability sum up to 1, so “normalize”. Division by broadcast, reshape the sum for each row as column, divide all columns for the same row by the same sum value. New sum of normalized ar(1) for each row is now equal to 1.

```
# current row sums
ar_row_sums_ar1 = np.sum(mt_ar1_trans, axis=1)
ar_row_sums_ar1 = np.reshape(ar_row_sums_ar1, [-1, 1])
# Update row values
mt_ar1_trans_norm = mt_ar1_trans/np.reshape(ar_row_sums_ar1, [-1, 1])
ar_row_sums_ar1_norm = np.sum(mt_ar1_trans_norm, axis=1)
ar_row_sums_ar1_norm = np.reshape(ar_row_sums_ar1_norm, [-1, 1])
# check sum
print(f'{np.column_stack([ar_row_sums_ar1, ar_row_sums_ar1_norm])=}')

## np.column_stack([ar_row_sums_ar1, ar_row_sums_ar1_norm])=array([[1.      , 1.      ],
##      [0.9999, 1.      ],
##      [1.      , 1.      ],
##      [1.0001, 1.      ],
##      [1.      , 1.      ],
##      [1.      , 1.      ],
##      [1.      , 1.      ],
##      [0.9999, 1.      ],
##      [0.9998, 1.      ],
##      [1.      , 1.      ],
##      [1.      , 1.      ],
##      [1.      , 1.      ],
##      [1.      , 1.      ],
##      [0.9999, 1.      ],
##      [1.0001, 1.      ],
##      [1.      , 1.      ],
##      [1.      , 1.      ],
##      [1.0001, 1.      ],
##      [1.      , 1.      ],
##      [1.      , 1.      ],
##      [1.0001, 1.      ],
##      [1.      , 1.      ],
##      [0.9998, 1.      ],
##      [1.      , 1.      ],
##      [1.      , 1.      ],
##      [0.9999, 1.      ],
##      [1.      , 1.      ],
##      [1.      , 1.      ],
##      [0.9999, 1.      ],
##      [1.0001, 1.      ],
##      [0.9998, 1.      ],
##      [0.9999, 1.      ],
##      [0.9999, 1.      ],
##      [1.      , 1.      ],
##      [1.      , 1.      ],
##      [1.0001, 1.      ],
##      [0.9998, 1.      ],
##      [1.0001, 1.      ],
##      [1.      , 1.      ],
##      [1.      , 1.      ],
##      [0.9999, 1.      ],
```

```
##      [0.9999, 1.    ],
##      [0.9999, 1.    ],
##      [1.0002, 1.    ],
##      [1.0002, 1.    ],
##      [0.9998, 1.    ],
##      [1.    , 1.    ],
##      [0.9999, 1.    ],
##      [0.9999, 1.    ],
##      [0.9999, 1.    ],
##      [0.9999, 1.    ],
##      [1.    , 1.    ]])
```