

# Python Pandas Generate Dataframes with Random Numeric and String Data

Fan Wang

2020-10-18

## Contents

<b>1</b>	<b>Generate Matrix from Arrays</b>	<b>1</b>
1.1	Single Arrays to Matrix . . . . .	1
1.2	Generate a Testing Dataframe with String and Numeric Values . . . . .	1

## 1 Generate Matrix from Arrays

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

```
import numpy as np
import pandas as pd
import random as random
import string as string
```

### 1.1 Single Arrays to Matrix

Given various arrays, generate a matrix

```
np.random.seed(123)
# Concatenate to matrix
mt_abc = np.column_stack(np.random.randint(10, size=(5, 3)))
# Matrix to data frame with columns and row names
df_abc = pd.DataFrame(data=mt_abc,
                      index=[ 'r' + str(it_col) for it_col in np.array(range(1, mt_abc.shape[0]+1))],
                      columns=[ 'c' + str(it_col) for it_col in np.array(range(1, mt_abc.shape[1]+1))])
# Print
print(df_abc)
```

```
##      c1  c2  c3  c4  c5
## r1    2   1   6   1   0
## r2    2   3   1   9   9
## r3    6   9   0   0   3
```

### 1.2 Generate a Testing Dataframe with String and Numeric Values

Generate a test dataframe with string and numeric variables. For testing purposes.

```
# Seed
np.random.seed(456)
random.seed(456)
```

```

# Numeric matrix 3 rows 4 columns
mt_numeric = np.random.randint(10, size=(3, 4))

# String block 5 letters per word, 3 rows and 3 columns of words
st_rand_word_block = ''.join(random.choice(string.ascii_lowercase) for ctr in range(5*3*3))
ls_st_rand_word = [st_rand_word_block[ctr: ctr + 5].capitalize() for ctr in range(0, len(st_rand_word_block), 5)]
mt_string = np.reshape(ls_st_rand_word, [3,3])

# Combine string and numeric matrix
mt_data = np.column_stack([mt_numeric, mt_string])

# Matrix to dataframe
df_data = pd.DataFrame(data=mt_data,
                        index=[ 'r' + str(it_col) for it_col in np.array(range(1, mt_data.shape[0]+1))],
                        columns=[ 'c' + str(it_col) for it_col in np.array(range(1, mt_data.shape[1]+1))])

# Print table
print(df_data)

```

```

##      c1 c2 c3 c4      c5      c6      c7
## r1  5  9  4  5  Xoonm  Zubtx  Zqdkp
## r2  7  1  8  3  Ydcpw  Obiee  Gfxmq
## r3  5  2  4  2  Tzrwu  Srwvp  Kcsrb

```