

to highlight
and positive

das and numpy
and generating

generates dataframe
filled with random

is a function
has in black.

of random values.

" Aim:

To create a database of ten rows, four columns with random values convert some values to nan values to highlight the nan values.

Pseudo code:

- * Import libraries: import pandas as np, numpy.
- * Create a dataframe, generate a database with 10 rows.
- * Convert values to nan, replace some of the values in the dataframe with np.nan to simulate missing data.
- * highlight Nan values.
- * Apply the style.

Sample input

Sample output:

A	B	C	D
0.8332	1.4076	0.060500	-0.326
-0.7050	1.594	1.099	-1.43
-6.505	-0.533	0.068	-0.898
-0.523	-6.017	-0.771	-0.415
-0.995	-1.778	-1.903	0.2949

Result.

This code is executed successfully and got the output.

```

import pandas as pd
import numpy as np

# Create a DataFrame with random values
data = np.random.randn(10, 4) # 10 rows, 4 columns
df = pd.DataFrame(data, columns=['A', 'B', 'C', 'D'])

# Introduce NaN values at random positions
nan_indices = [(0, 1), (2, 2), (4, 0), (6, 3), (9, 2)] # List of indices where NaNs will be introduced
for idx in nan_indices:
    df.iloc[idx] = np.nan

# Highlight NaN values using style
def highlight_nan(val):
    color = 'red' if pd.isna(val) else ''
    return f'background-color: {color}'

# Apply the styling
styled_df = df.style.applymap(highlight_nan)

# Display the styled DataFrame
styled_df

```

<ipython-input-2-4d34c7922de0>:19: FutureWarning: Styler.applymap has been deprecated. Use Styler.map instead.

```

styled_df = df.style.applymap(highlight_nan)

```

	A	B	C	D
0	0.363316	nan	0.594925	0.968790
1	0.258651	1.199258	-1.479719	-0.113387
2	0.295008	-0.134901	nan	-1.298162
3	-0.365289	-0.084684	1.290258	-1.200692
4	nan	-0.356519	0.294632	-0.136161
5	-1.795682	0.292742	-0.163703	1.205948