

## Experiment - 11

AIM: To create dataframe of ten rows, four columns with random values convert some values to nan values.

Pseudocode:

- 1) Import pandas & numpy.
- 2) create a dataframe with 10 rows & 4 columns.
- 3) Replace some of the values in the dataframe with np.nan to simulate missing data.
- 4) Highlight NaN values.
- 5) Apply the style.

Sample Input:

	A	B	C	D
0	0.71			
1	0.32	NAN	0.65	0.12
2	NAN	0.45	NAN	0.89
3		0.14	0.76	0.31
4	0.27	0.43	0.69	NAN

Sample Output:

	A	B	C	D
0	NAN	0.6438	0.4882	NAN
1	0.8046	0.009066	0.38199	0.563411
2	0.7767	0.0412	0.1141	0.39587

Result:

The code is executed successfully & got

```

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