-0.543	-1.017	-0.771	-0.415
-0.995	-1.778	-1.903	0.2949

Result:-

This Code is excuited successfully and go the output.

[illegible]

11. Create a dataframe of ten rows, four columns with random Values. convert some Values to nan Values. write a pandas program which will highlight the nan Value.

Aim:-

To Create a Dataframe with random Values and introduces Some NaN (missing) Values. the goal is to highlight these NaN Values using conditional formating in pandas.

Pesudo Code:

```
⇒ Import the pandas and numpy libiraries.
⇒ Generate a Dataframe with random Values and set
dimension as 10 rows × 4 Columns.
```

coursefaculty

8/11/2024.

mentor/HOD.

→ Randomly assign some entries as NaN using numpy function

⇒ Create a function that applies a specific style for NaN Values

⇒ Display the style to Dataframe.

Sample Input

	A	B	C	D
0	0.78	NaN	0.65	0.12
1	0.32	0.45	NaN	0.89
2	NaN	0.14	0.76	0.37
....	....	....	....	....
9	0.29	0.43	0.69	NaN

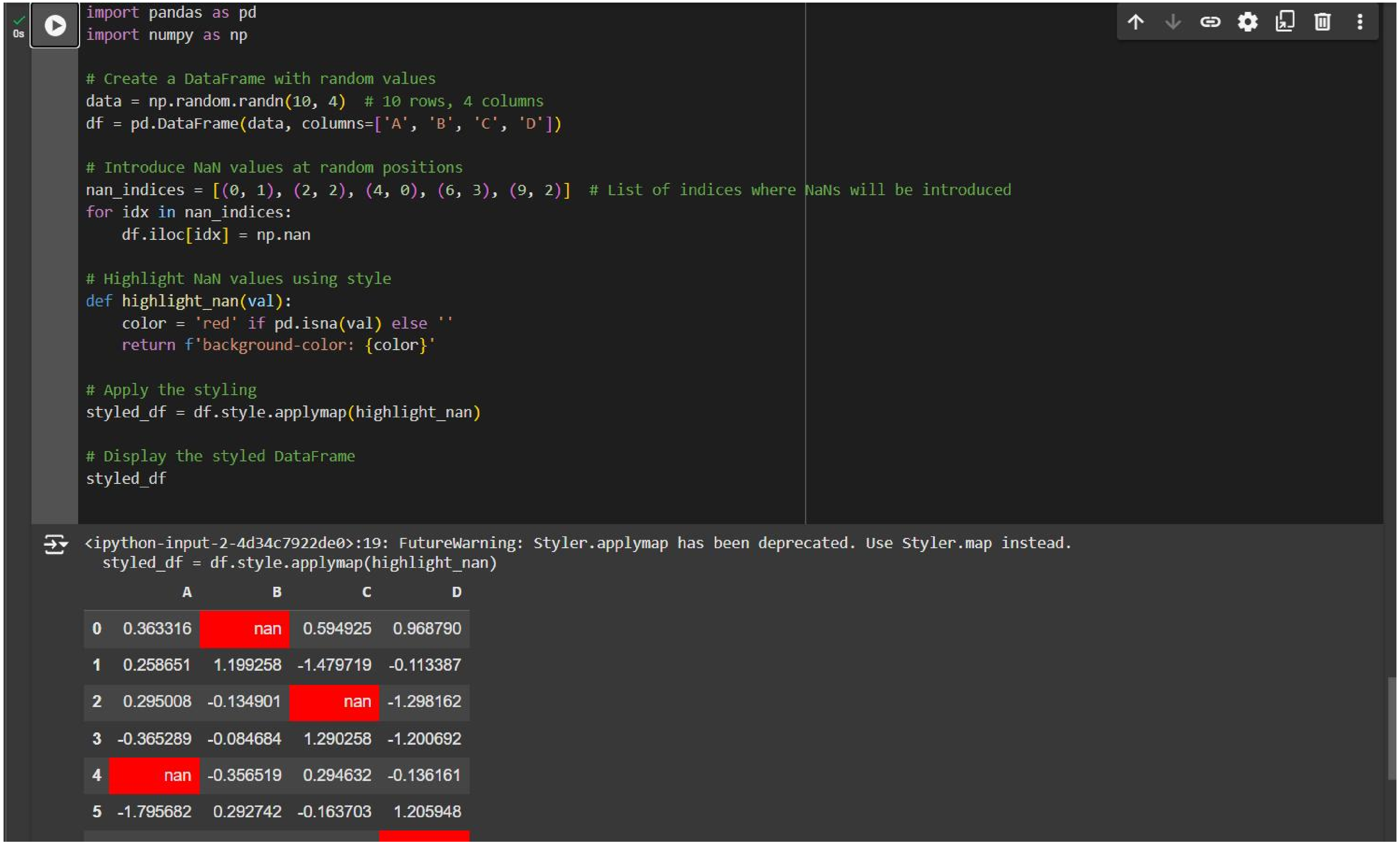
Sample output

	A	B	C	D
0	nan	0.693894	0.488209	nan
1	0.804680	0.009066	0.384993	0.563411
2	0.547820	0.940590	0.811418	0.87544
....	....	....	....	....
9	0.776698	0.091820	0.085177	0.168030

Result:

This Code is excuted Successfully and go the output.

[illegible]



```python
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
```