

```

In [2]: import pandas as pd
import matplotlib.pyplot as plt
data_path = 'C:\\Users\\DELL\\Downloads\\archive (4)\\website_wata.csv'
df = pd.read_csv(data_path)
print(df.head())
df['Conversion Rate'].isnull().sum()

df['Traffic Source'].unique()

import pandas as pd
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df['Traffic Source'] = le.fit_transform(df['Traffic Source'])

import pandas as pd
import statsmodels.api as sm
import matplotlib.pyplot as plt

plt.figure(figsize=(15, 10))
plt.subplot(3, 2, 1)
df['Page Views'].plot(title='Page Views')
plt.subplot(3, 2, 2)
df['Session Duration'].plot(title='Session Duration')
plt.subplot(3, 2, 3)
df['Bounce Rate'].plot(title='Bounce Rate')
plt.subplot(3, 2, 4)
df['Time on Page'].plot(title='Time on Page')
plt.subplot(3, 2, 5)
df['Previous Visits'].plot(title='Previous Visits')
plt.subplot(3, 2, 6)
df['Conversion Rate'].plot(title='Conversion Rate')
plt.tight_layout()
plt.show()

df['Date'] = pd.date_range(start='2023-01-01', periods=df.shape[0], freq='D')
df.set_index('Date', inplace=True)

df['2023-01']

decomposition_page_views = sm.tsa.seasonal_decompose(df['Page Views'], model='additive')
decomposition_session_duration = sm.tsa.seasonal_decompose(df['Session Duration'], model='additive')

trend_page_views = decomposition_page_views.trend
seasonal_page_views = decomposition_page_views.seasonal
residual_page_views = decomposition_page_views.resid

plt.figure(figsize=(10, 8))
plt.subplot(411)
plt.plot(df['Page Views'], label='Original')
plt.legend(loc='best')
plt.title('Page Views')

plt.subplot(412)
plt.plot(trend_page_views, label='Trend')
plt.legend(loc='best')
plt.title('Trend Component')

```

```

plt.subplot(413)
plt.plot(seasonal_page_views, label='Seasonal')
plt.legend(loc='best')
plt.title('Seasonal Component')

plt.subplot(414)
plt.plot(residual_page_views, label='Residual')
plt.legend(loc='best')
plt.title('Residual Component')

plt.tight_layout()
plt.show()

df["Page Views"].resample('M').mean().plot()

df["Traffic Source"].resample('M').mean().plot()

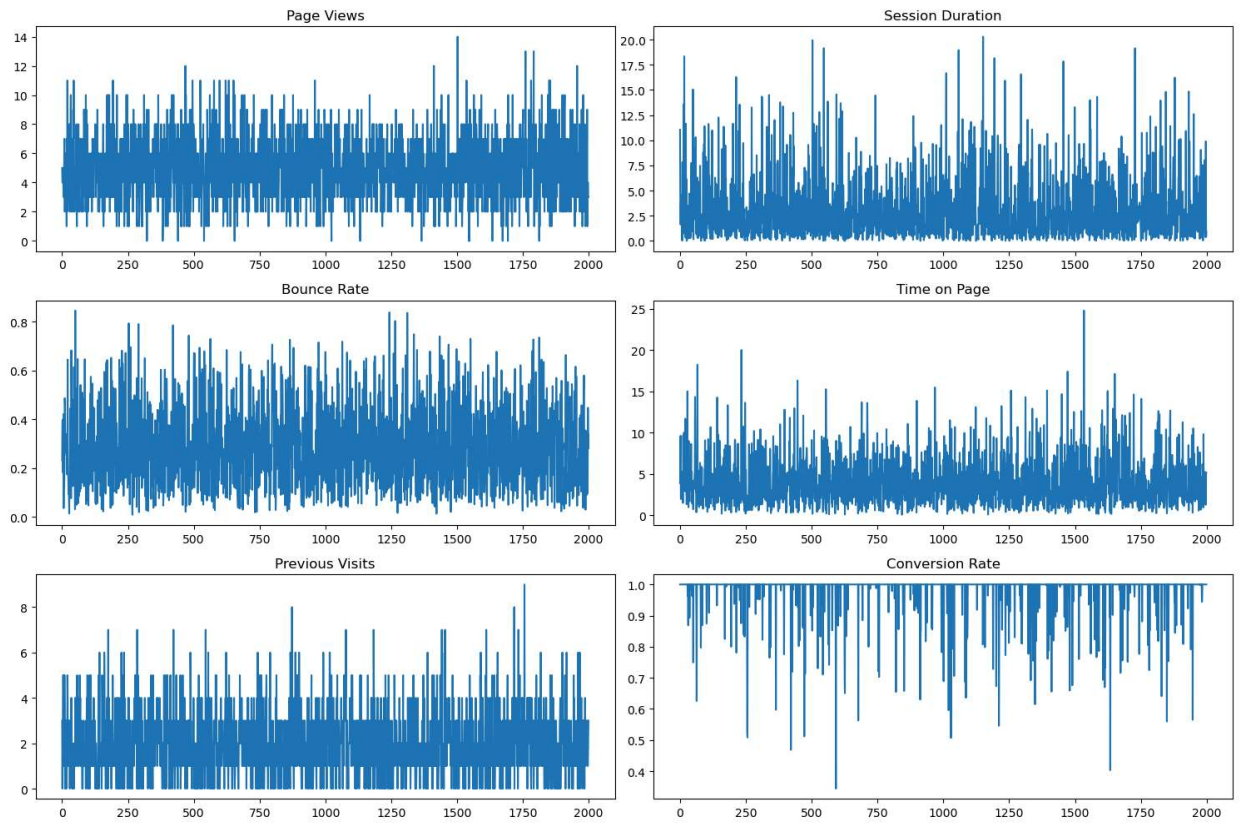
df["Bounce Rate"].resample('M').mean().plot()

window_size = 15
moving_average = df['Session Duration'].rolling(window=window_size).mean()
plt.figure(figsize=(12, 6))
plt.plot(df['Session Duration'], label='Original')
plt.plot(moving_average, label='Moving Average')
plt.legend()
plt.title('Original Time Series vs. Moving Average')
plt.show()

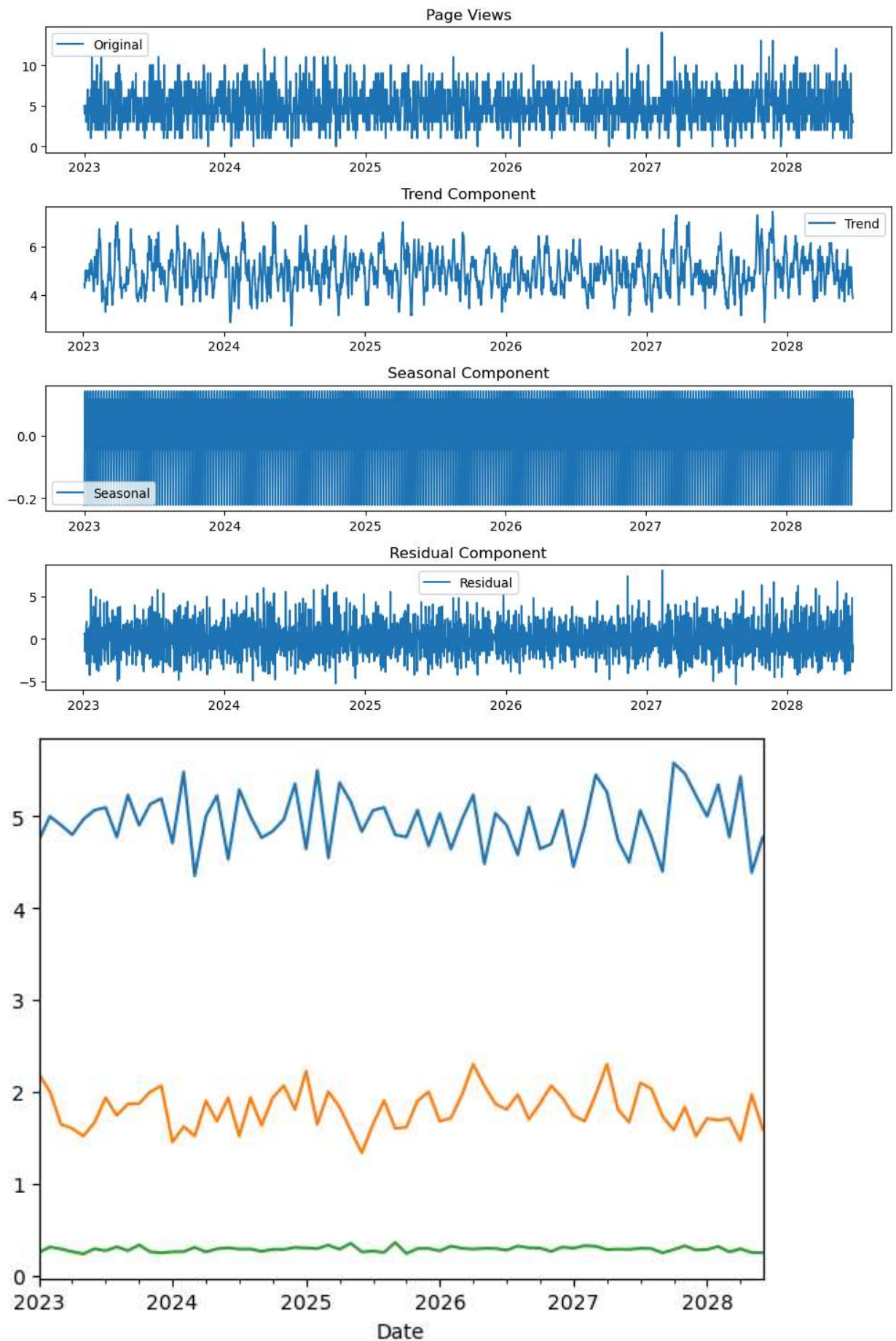
```

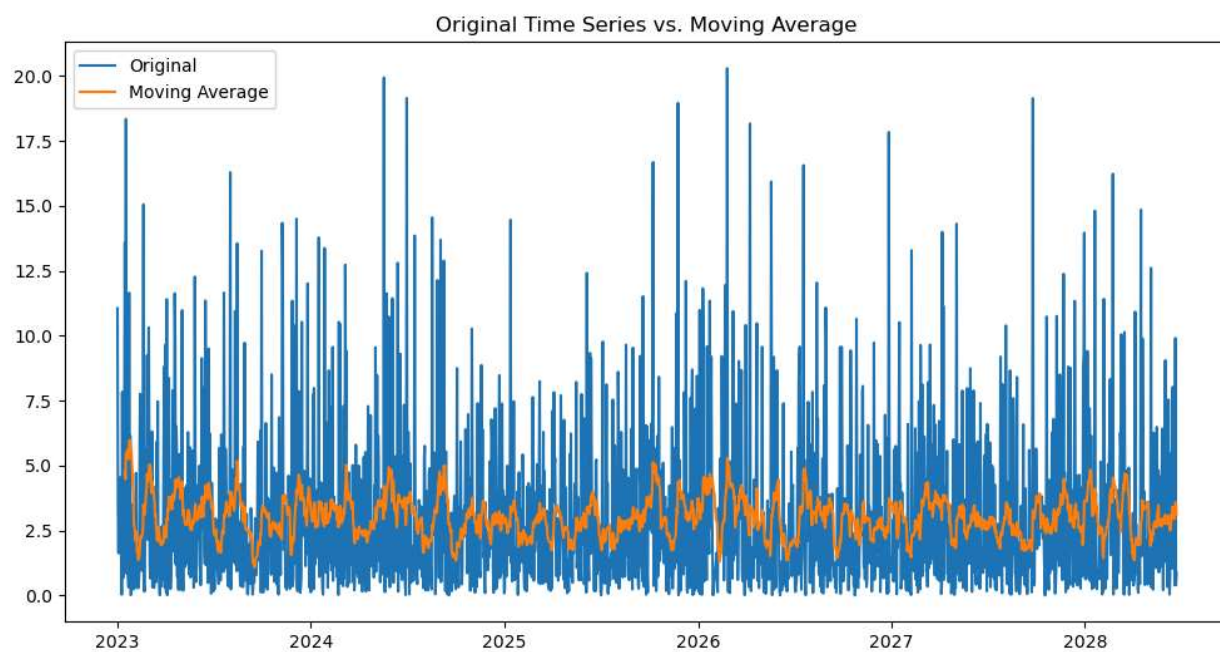
	Page Views	Session Duration	Bounce Rate	Traffic Source	Time on Page \
0	5	11.051381	0.230652	Organic	3.890460
1	4	3.429316	0.391001	Social	8.478174
2	4	1.621052	0.397986	Organic	9.636170
3	5	3.629279	0.180458	Organic	2.071925
4	5	4.235843	0.291541	Paid	1.960654

	Previous Visits	Conversion Rate
0	3	1.0
1	0	1.0
2	2	1.0
3	3	1.0
4	5	1.0



```
C:\Users\DELL\AppData\Local\Temp\ipykernel_3584\3445912530.py:39: FutureWarning: Indexing a DataFrame with a datetimelike index using a single string to slice the rows, like `frame[string]`, is deprecated and will be removed in a future version. Use `frame.loc[string]` instead.  
df['2023-01']
```





In []: