Coding Challenge

Web Data Analysis

Name: Shreyas Wani Date: 15/11/2024

Q1. The team wants to analyze each variable of the data collected through data summarization to get a basic understanding of the dataset and to prepare for further analysis.

```
data
```

	Bounces	Exits	Continent	Sourcegroup	Timeinpage	Uniquepageviews	Visits	BouncesNew
0	0	0	OC	(direct)	18	1	0	0.00
1	0	0	N.America	(direct)	4	1	0	0.00
2	0	0	N.America	Others	35	1	0	0.00
3	0	0	N.America	public.tableausoftware.com	70	1	0	0.00
4	0	0	N.America	public.tableausoftware.com	81	1	0	0.00
...	...	...	...	...	...	...	...	...
32104	1	1	N.America	public.tableausoftware.com	12	2	2	0.01
32105	2	2	N.America	(direct)	0	2	2	0.02
32106	2	2	N.America	(direct)	0	2	2	0.02
32107	2	2	N.America	(direct)	0	2	2	0.02
32108	2	2	N.America	google	0	2	2	0.02

Loading the dataset into a DataFrame for analysis

```
data.describe()
```

	Bounces	Exits	Timeinpage	Uniquepageviews	Visits	BouncesNew
count	32109.000000	32109.000000	32109.000000	32109.000000	32109.000000	32109.000000
mean	0.713009	0.906039	73.184746	1.114329	0.906039	0.007130
std	0.708215	0.695819	394.441111	0.614880	0.730068	0.007082
min	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000
25%	0.000000	1.000000	0.000000	1.000000	1.000000	0.000000
50%	1.000000	1.000000	0.000000	1.000000	1.000000	0.010000
75%	1.000000	1.000000	10.000000	1.000000	1.000000	0.010000
max	30.000000	36.000000	46745.000000	45.000000	45.000000	0.300000

Summarizes statistical details like mean, median, and standard deviation for numerical columns.

```
data.dtypes
```

```
Bounces               int64
Exits                 int64
Continent            object
Sourcegroup          object
Timeinpage            int64
Uniquepageviews       int64
Visits                int64
BouncesNew          float64
dtype: object
```

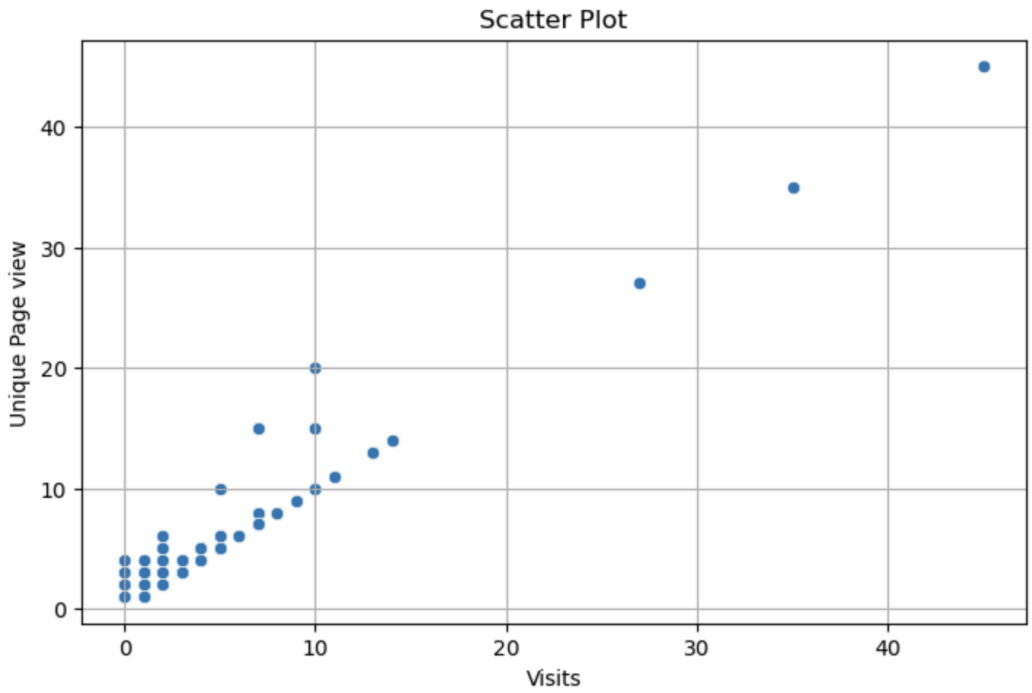
Checks the data types of columns to ensure compatibility for analysis and visualization.

Q2. As mentioned earlier, a unique page view represents the number of sessions during which that page was viewed one or more times. A visit counts all instances, no matter how many times the same visitor may have been to your site. So the team needs to know whether the unique page view value depends on visits.

```
correlation = data['Visits'].corr(data['Uniquepageviews'])
print(f"Correlation Coefficient: {correlation:.2f}")
```

```
Correlation Coefficient: 0.81
```

```
plt.figure(figsize=(8, 5))
sns.scatterplot(x='Visits', y='Uniquepageviews', data=data)
plt.title("Scatter Plot")
plt.xlabel("Visits")
plt.ylabel("Unique Page view")
plt.grid()
plt.show()
```



Calculates and displays the correlation strength and direction between **Visits** and **Unique Page Views**. Displays the relationship between **Visits** and **Unique Page Views** to identify trends or patterns

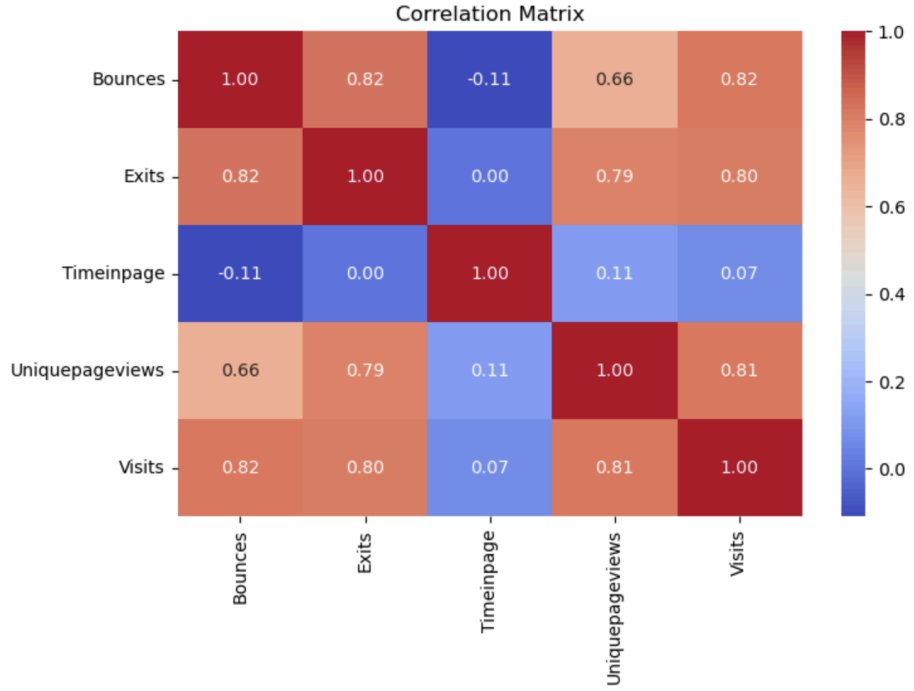
Q3. Find out the probable factors from the dataset, which could affect the exits. Exit Page Analysis is usually required to get an idea about why a user leaves the website for a session and moves on to another one. Please keep in mind that exits should not be confused with bounces.

```
vairables =['Exits', 'Timeinpage', 'Bounces','Uniquepageviews', 'Visits']
data[variables].corr()
```

	Bounces	Exits	Timeinpage	Uniquepageviews	Visits
Bounces	1.000000	0.824912	-0.109106	0.659101	0.819343
Exits	0.824912	1.000000	0.001325	0.791129	0.800979
Timeinpage	-0.109106	0.001325	1.000000	0.114593	0.066650
Uniquepageviews	0.659101	0.791129	0.114593	1.000000	0.814446
Visits	0.819343	0.800979	0.066650	0.814446	1.000000

```
plt.figure(figsize=(8, 5))
sns.heatmap(data[variables].corr(), annot=True, fmt=".2f", cmap="coolwarm")
plt.title("Correlation Matrix")
plt.show()

correlation = data.corr()['Exits'].sort_values(ascending=False)
print("\nCorrelation with Exits:")
print(correlation)
```



```
correlation = data.corr()['Exits'].sort_values(ascending=False)
print("\nCorrelation with Exits:")
print(correlation)
```

```
Correlation with Exits:
Exits              1.000000
BouncesNew         0.824912
Bounces            0.824912
Visits             0.800979
Uniquepageviews    0.791129
Timeinpage         0.001325
Name: Exits, dtype: float64
```

Computes correlations among variables like **Exits**, **Bounces**, and **Visits** to assess interdependencies.
Visualizes correlations among selected variables to quickly identify strong or weak relationships.
Identifies variables that have a strong positive or negative correlation with **Exits**.

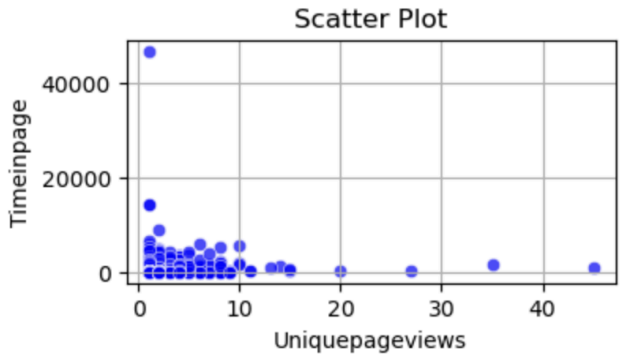
Q4. Every site wants to increase the time on page for a visitor. This increases the chances of the visitor understanding the site content better and hence there are more chances of a transaction taking place. Find the variables which possibly have an effect on the time on page.

```
time_correlation = data.corr()['Timeinpage'].sort_values(ascending=False)
print("\nCorrelation with Time on Page:")
print(time_correlation)
```

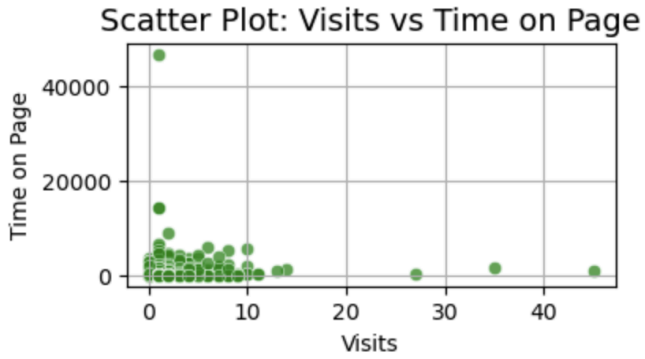
```
Correlation with Time on Page:
Timeinpage         1.000000
Uniquepageviews    0.114593
Visits             0.066650
Exits              0.001325
Bounces           -0.109106
BouncesNew        -0.109106
Name: Timeinpage, dtype: float64
```

Highlights factors that influence **Time on Page** by showing their correlation values.

```
plt.figure(figsize=(4, 2))
sns.scatterplot(x='Uniquepageviews', y='Timeinpage', data=data, color="blue", alpha=0.7)
plt.title("Scatter Plot")
plt.xlabel("Uniquepageviews")
plt.ylabel("Timeinpage")
plt.grid()
plt.show()
```




```
    plt.figure(figsize=(4, 2))
    sns.scatterplot(x='Visits', y='Timeinpage', data=data, color="green", alpha=0.7)
    plt.title("Scatter Plot: Visits vs Time on Page", fontsize=14)
    plt.xlabel("Visits")
    plt.ylabel("Time on Page")
    plt.grid()
    plt.show()
```

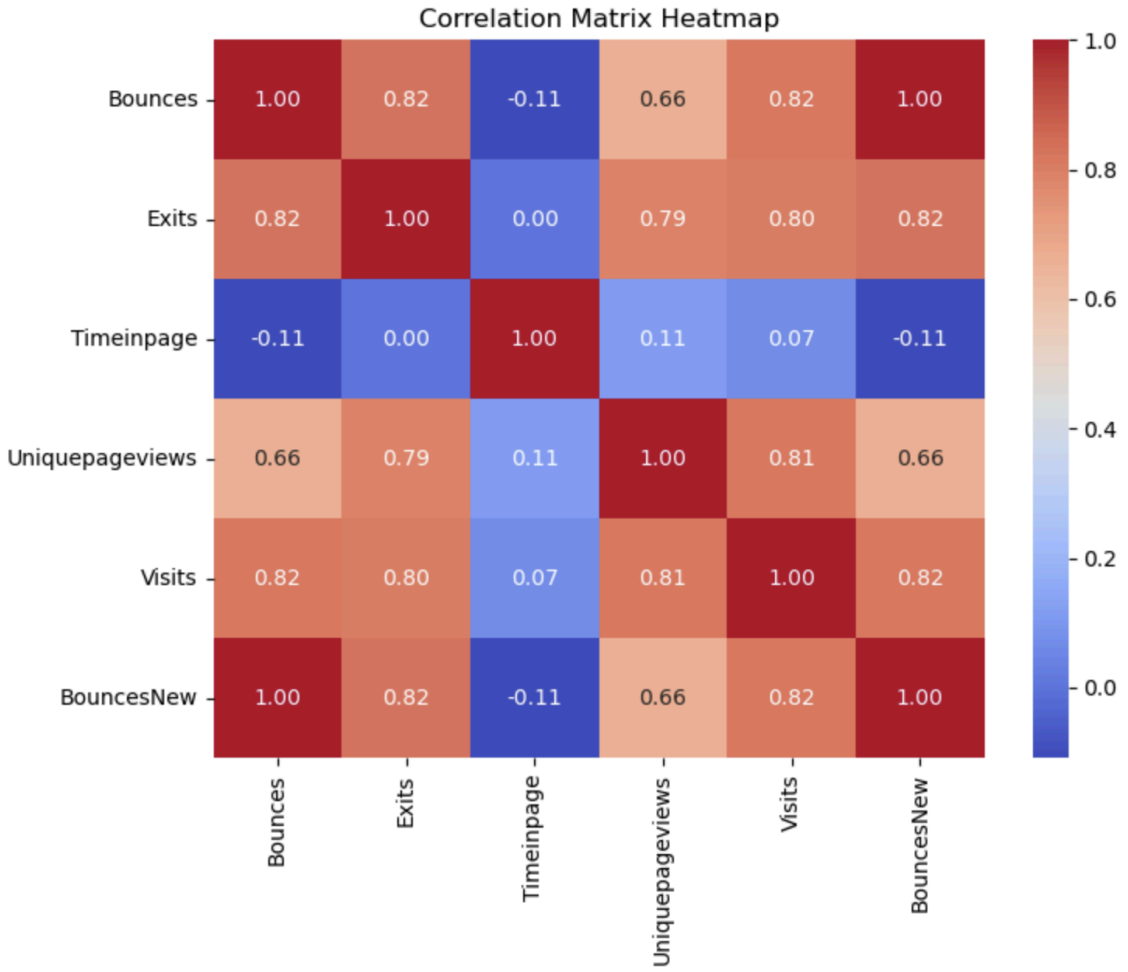


Explores how Unique Page Views relate to Time on Page, identifying possible trends.
Analyzes the impact of increased Visits on Time on Page using a visual plot

Displays a heatmap for correlations across all numerical variables in the dataset.

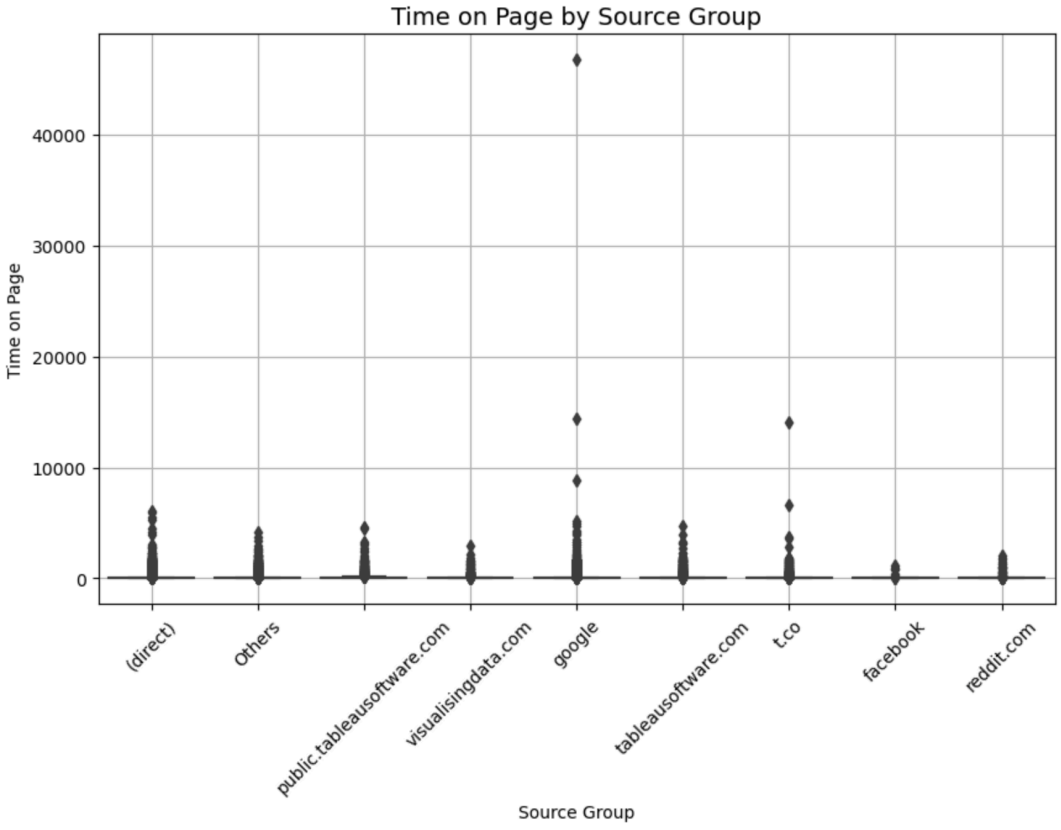
```
plt.figure(figsize=(8, 6))
sns.heatmap(data.corr(), annot=True, cmap='coolwarm', fmt='.2f', cbar=True)
plt.title('Correlation Matrix Heatmap')
plt.show()
```

```
/var/folders/l8/m03wlrfn6gb5r7l9hwz69rj40000gn/T/ipykernel_86273/755983166.py:2: FutureWarning
f numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False
lumns or specify the value of numeric_only to silence this warning.
  sns.heatmap(data.corr(), annot=True, cmap='coolwarm', fmt='.2f', cbar=True)
```



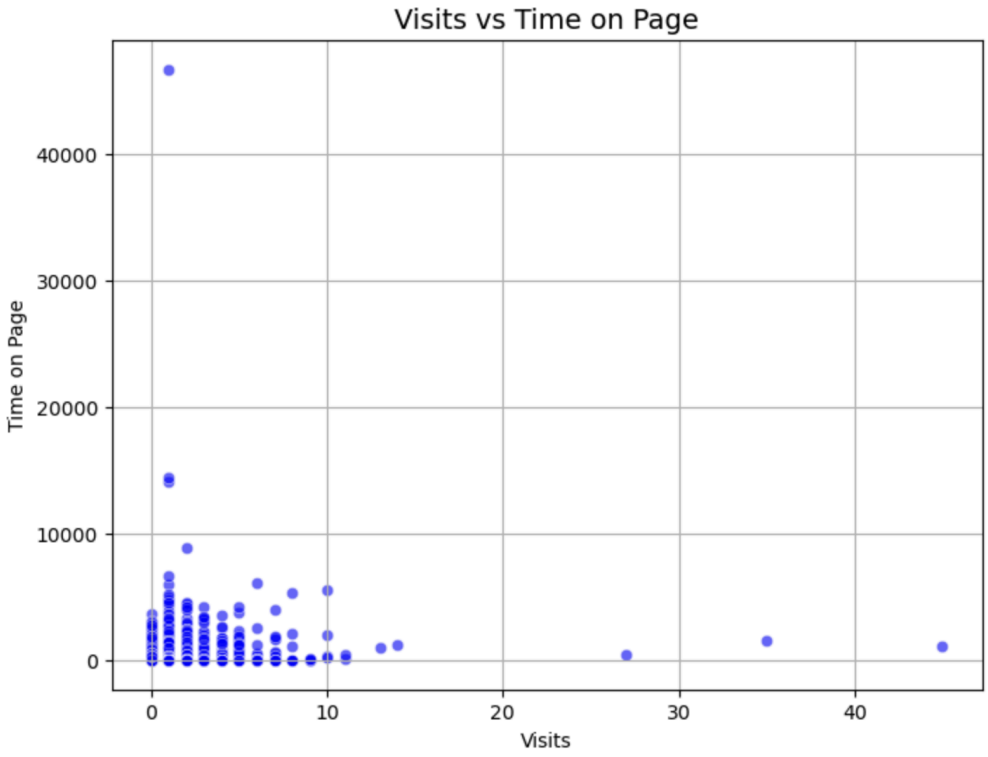
Compares **Time on Page** across different **Source Groups** to identify differences.

```
# Source Group is a categorical variable
plt.figure(figsize=(10, 6))
sns.boxplot(x='Sourcegroup', y='Timeinpage', data=data)
plt.title("Time on Page by Source Group", fontsize=14)
plt.xlabel("Source Group")
plt.ylabel("Time on Page")
plt.xticks(rotation=45)
plt.grid(True)
plt.show()
```



Further examines the link between **Visits** and **Time on Page** visually.

```
)]: # more Visits lead to more Time on Page
    plt.figure(figsize=(8, 6))
    sns.scatterplot(x='Visits', y='Timeinpage', data=data, color='blue', alpha=0.6)
    plt.title("Visits vs Time on Page", fontsize=14)
    plt.xlabel("Visits")
    plt.ylabel("Time on Page")
    plt.grid(True)
    plt.show()
```



Q5. A high bounce rate is a cause of alarm for websites which depend on visitor engagement. Help the team in determining the factors that are impacting the bounce.

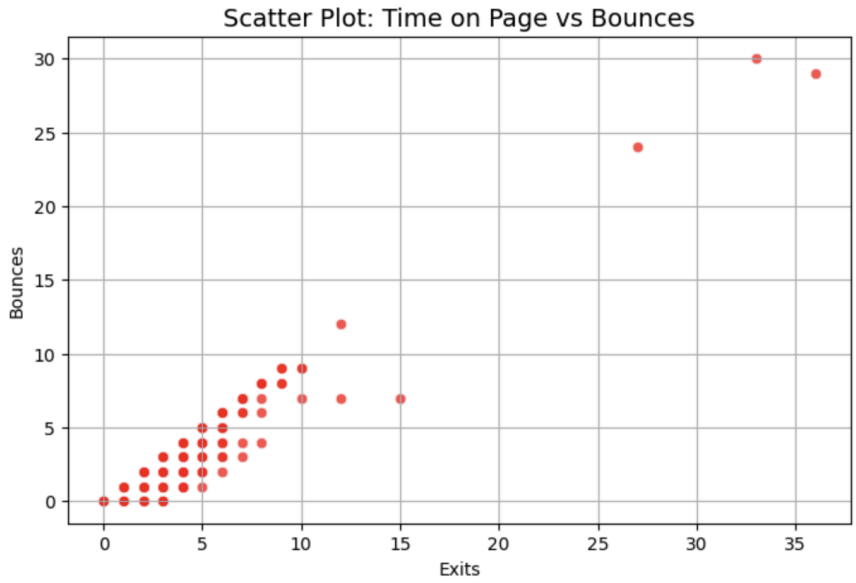
```
bounce_correlation = data.corr()['Bounces'].sort_values(ascending=False)
print("\nCorrelation with Bounces:")
print(bounce_correlation)
```

```
Correlation with Bounces:
Bounces             1.000000
BouncesNew          1.000000
Exits               0.824912
Visits              0.819343
Uniquepageviews     0.659101
Timeinpage         -0.109106
Name: Bounces, dtype: float64
```

Analyzes which variables correlate with **Bounces** to understand bounce behavior.

Shows the relationship between **Exits** and **Bounces**, highlighting possible dependencies.

```
plt.figure(figsize=(8, 5))
sns.scatterplot(x='Exits', y='Bounces', data=data, color="red", alpha=0.7)
plt.title("Scatter Plot: Time on Page vs Bounces", fontsize=14)
plt.xlabel("Exits")
plt.ylabel("Bounces")
plt.grid()
plt.show()
```



Examines how **Time on Page** impacts **Bounces**, providing insights into user engagement.

```
plt.figure(figsize=(8, 5))
sns.scatterplot(x='Timeinpage', y='Bounces', data=data, color="red", alpha=0.7)
plt.title("Scatter Plot: Time on Page vs Bounces", fontsize=14)
plt.xlabel("Time on Page")
plt.ylabel("Bounces")
plt.grid()
plt.show()
```