```python
import pandas as pd
```
Python

```python
# Load the dataset
data = pd.read_csv(r"C:/Users/SHRI/OneDrive/Desktop/Capstone Project/traffic/traffic.csv")
```
Python

```python
# Task 1: Total and Daily Pageview Events
pageview_data = data[data['event'] == 'pageview']
total_pageviews = pageview_data.shape[0]
```
Python

```python
# Ensure 'date' column is in datetime format
pageview_data['date'] = pd.to_datetime(pageview_data['date'], errors='coerce')

# Drop rows with invalid or missing 'date' values
pageview_data = pageview_data.dropna(subset=['date'])

# Group by 'date' and calculate daily pageviews
daily_pageviews = pageview_data.groupby('date').size()

# Calculate the average number of pageviews per day
average_pageviews_per_day = daily_pageviews.mean()
```
Python

```python
#Task 1: Total and Daily Pageview Events
print("Task 1: Total and Daily Pageview Events")
print(f"Total Pageviews: {total_pageviews}")
print("Daily Pageviews:")
print(daily_pageviews)
print(f"Average Pageviews Per Day: {average_pageviews_per_day}")
```
Python

```
Task 1: Total and Daily Pageview Events
Total Pageviews: 142015
Daily Pageviews:
date
2021-08-19    22366
2021-08-20    21382
2021-08-21    21349
2021-08-22    20430
2021-08-23    18646
2021-08-24    18693
2021-08-25    19149
dtype: int64
Average Pageviews Per Day: 20287.85714285714
```

```python
# Task 2: Analysis of Other Events
event_counts = data['event'].value_counts()
event_distribution = event_counts / event_counts.sum() * 100
print("\nTask 2: Event Analysis")
print("Event Counts:")
print(event_counts)
print("\nEvent Distribution (%):")
print(event_distribution)
```

Python

```
Task 2: Event Analysis
Event Counts:
event
pageview     142015
click         55732
preview       28531
Name: count, dtype: int64

Event Distribution (%):
event
pageview     62.761294
click        24.629880
preview      12.608826
Name: count, dtype: float64
```

```python
# Task 3: Geographical Distribution
geo_pageview_data = pageview_data.dropna(subset=['country'])
pageviews_by_country = geo_pageview_data['country'].value_counts()
print("\nTask 3: Geographical Distribution")
print("Pageviews by Country:")
print(pageviews_by_country.head(10))  # Display top 10 countries
```
Python

```
Task 3: Geographical Distribution
Pageviews by Country:
country
Saudi Arabia          28873
India                 27286
United States         20839
France                 9674
Iraq                   4897
United Kingdom         3845
Pakistan               3212
Germany                3141
Turkey                 2462
United Arab Emirates   2335
Name: count, dtype: int64
```

```python
# Task 4: Click-Through Rate (CTR) Analysis
click_data = data[data['event'] == 'click']
clicks_by_link = click_data['linkid'].value_counts()
pageviews_by_link = pageview_data['linkid'].value_counts()
ctr_by_link = (clicks_by_link / pageviews_by_link).fillna(0) * 100
overall_ctr = (click_data.shape[0] / total_pageviews) * 100
print("\nTask 4: Click-Through Rate (CTR) Analysis")
print(f"Overall CTR: {overall_ctr:.2f}%")
print("CTR by Link (Top 10):")
print(ctr_by_link.head(10))
```

Python

```
Task 4: Click-Through Rate (CTR) Analysis
Overall CTR: 39.24%
CTR by Link (Top 10):
linkid
00073307-ae96-5089-a117-4783afb42f8e      0.000000
00126b32-0c35-507b-981c-02c80d2aa8e7    100.000000
0018cfff-50a1-5984-9715-01ef2d11a49a      0.000000
0033934b-5d16-5a06-af58-d087bcdd3680      0.000000
0034d6cf-3bd8-5ffe-aafc-b3959fc48608      0.000000
0038ff20-1ef7-5846-9f1a-4012fdf24da7      0.000000
004b9724-abca-5481-b6e9-6148a7ca00a5    100.000000
0062d109-f6f4-5637-8731-8ce38362c113      0.000000
00633299-125d-5d5c-a081-3b2cc4d89ffd      0.000000
0063a982-41cd-5629-96d0-e1c4dd72ea11     66.666667
Name: count, dtype: float64
```

```python
# Task 5: Correlation Analysis
from scipy.stats import pearsonr, spearmanr
```
Python

```python
# Prepare data for correlation
clicks_and_pageviews = pd.DataFrame({
    "clicks": click_data['linkid'].value_counts(),
    "pageviews": pageview_data['linkid'].value_counts()
}).fillna(0)
```
Python

```python
# Pearson correlation (linear relationship)
pearson_corr, pearson_pval = pearsonr(clicks_and_pageviews['clicks'], clicks_and_pageviews['pageviews'])
```
Python

```python
# Spearman correlation (rank-based relationship)
spearman_corr, spearman_pval = spearmanr(clicks_and_pageviews['clicks'], clicks_and_pageviews['pageviews'])
```
Python

```python
print("\nTask 5: Correlation Analysis")
print(f"Pearson Correlation: {pearson_corr:.2f} (p-value: {pearson_pval:.4f})")
print(f"Spearman Correlation: {spearman_corr:.2f} (p-value: {spearman_pval:.4f})")
```

Python

```
Task 5: Correlation Analysis
Pearson Correlation: 0.99 (p-value: 0.0000)
Spearman Correlation: 0.63 (p-value: 0.0000)
```