



WEBSITE TRAFFIC ANALYSIS

Performing Different analysis and Visualizatipon using IBM
Cognos


AGENDA

- Introduction
- Processing techniques
- Visualization using IBM cognos
- conclusion





INTRODUCTION

- 
- A website traffic analysis project involves the collection, processing, and analysis of data related to the interactions and behaviors of users on a website. This type of project aims to provide insights into how visitors are engaging with a website, allowing website owners and administrators to make informed decisions for optimizing user experience and achieving specific business goals.



+ Code + Text



✓
1s



```
import pandas as pd
import numpy as np
import random
from datetime import datetime, timedelta

# Generate sample dates
start_date = datetime(2023, 1, 1)
end_date = datetime(2023, 12, 31)
date_range = [start_date + timedelta(days=i) for i in range((end_date - start_date).days)]

# Create a list of random website pages
pages = ["home", "about", "products", "blog", "contact"]

# Generate random data
data = {
    "Date": [random.choice(date_range) for _ in range(1000)],
    "Page": [random.choice(pages) for _ in range(1000)],
    "Visits": np.random.randint(1, 100, size=1000),
}

# Create a Pandas DataFrame
df = pd.DataFrame(data)

# Save the dataset to a CSV file
df.to_csv("website_traffic_dataset.csv", index=False)

print("Website traffic dataset created and saved as website_traffic_dataset.csv")
import pandas as pd
```




✓ 1s completed at 7:29 AM





CONCLUSION

- 
- In conclusion, the Website Traffic Analysis Project serves as a fundamental component of website management and digital marketing strategies. By collecting, processing, and analyzing user data, this project provides invaluable insights into how visitors interact with a website. The outcomes of such analysis empower website owners and administrators to make informed decisions for optimizing user experiences, content strategies, conversion rates, search engine rankings, and marketing effectiveness.

```
# Generate sample dates
start_date = datetime(2023, 1, 1)
end_date = datetime(2023, 12, 31)
date_range = [start_date + timedelta(days=i) for i in range((end_date - start_date).days)]

# Create a list of random website pages
pages = ["home", "about", "products", "blog", "contact"]

# Generate random data
data = {
    "Date": [random.choice(date_range) for _ in range(1000)],
    "Page": [random.choice(pages) for _ in range(1000)],
    "Visits": np.random.randint(1, 100, size=1000),
}

# Create a Pandas DataFrame
df = pd.DataFrame(data)

# Save the dataset to a CSV file
df.to_csv("website_traffic_dataset.csv", index=False)

print("Website traffic dataset created and saved as website_traffic_dataset.csv")
import pandas as pd
import numpy as np
import random
from datetime import datetime, timedelta

# Generate sample dates
start_date = datetime(2023, 1, 1)
```

+ Code + Text

RAM
Disk

```
# Generate sample dates
start_date = datetime(2023, 1, 1)
end_date = datetime(2023, 12, 31)
date_range = [start_date + timedelta(days=i) for i in range((end_date - start_date).days)]

# Create a list of random website pages
pages = ["home", "about", "products", "blog", "contact"]

# Generate random data
data = {
    "Date": [random.choice(date_range) for _ in range(1000)],
    "Page": [random.choice(pages) for _ in range(1000)],
    "Visits": np.random.randint(1, 100, size=1000),
}

# Create a Pandas DataFrame
df = pd.DataFrame(data)

# Save the dataset to a CSV file
df.to_csv("website_traffic_dataset.csv", index=False)

print("Website traffic dataset created and saved as website_traffic_dataset.csv")
import pandas as pd
import numpy as np
import random
from datetime import datetime, timedelta

# Generate sample dates
start_date = datetime(2023, 1, 1)
```

```
[1] df.to_csv("website_traffic_dataset.csv", index=False)

print("Website traffic dataset created and saved as website_traffic_dataset.csv")
```

Website traffic dataset created and saved as website_traffic_dataset.csv
Website traffic dataset created and saved as website_traffic_dataset.csv
Website traffic dataset created and saved as website_traffic_dataset.csv
Website traffic dataset created and saved as website_traffic_dataset.csv
Website traffic dataset created and saved as website_traffic_dataset.csv
Website traffic dataset created and saved as website_traffic_dataset.csv
Website traffic dataset created and saved as website_traffic_dataset.csv

```
import pandas as pd

# Load the dataset from the CSV file
df = pd.read_csv("website_traffic_dataset.csv")

# Display the first few rows of the dataset
print(df.head())
```

	Date	Page	Visits
0	2023-08-07	about	39
1	2023-05-31	products	38
2	2023-10-16	products	65
3	2023-12-22	about	60
4	2023-10-27	blog	71


```
import pandas as pd

# Load the dataset from the CSV file
df = pd.read_csv("website_traffic_dataset.csv")

# Display the first few rows of the dataset
print("Original Dataset:")
print(df.head())

# Data Processing Techniques:

# 1. Filter data for a specific page
page = "home"
filtered_df = df[df["Page"] == page]
print("\nData for the '{}' page:".format(page))
print(filtered_df.head())

# 2. Group data by date and calculate daily total visits
daily_visits = df.groupby("Date")["Visits"].sum().reset_index()
print("\nDaily Total Visits:")
print(daily_visits.head())

# 4. Find the most visited page
most_visited_page = df["Page"].value_counts().idxmax()
print("\nMost Visited Page:", most_visited_page)

# 5. Sort the data by date in ascending order
```

```
# import matplotlib.pyplot as
```

Original Dataset:

	Date	Page	Visits
0	2023-08-07	about	39
1	2023-05-31	products	38
2	2023-10-16	products	65
3	2023-12-22	about	60
4	2023-10-27	blog	71

Data for the 'home' page:

	Date	Page	Visits
6	2023-05-21	home	18
11	2023-09-29	home	68
27	2023-02-03	home	75
28	2023-10-11	home	22
30	2023-12-08	home	46

Daily Total Visits:

	Date	Visits
0	2023-01-01	215
1	2023-01-02	143
2	2023-01-03	42
3	2023-01-05	184
4	2023-01-06	91

Most Visited Page: products

Sorted Dataset by Date:

	Date	Page	Visits
704	2023-01-01	blog	82

```
+ Code + Text
[6] 4 2023-01-06 91

Most Visited Page: products

Sorted Dataset by Date:
      Date    Page  Visits
704 2023-01-01  blog    82
403 2023-01-01  blog    71
156 2023-01-01 contact   62
641 2023-01-02 contact   54
447 2023-01-02  blog    89

Summary Statistics:
      Visits
count 1000.000000
mean   48.908000
std    27.965865
min     1.000000
25%    25.000000
50%    47.000000
75%    72.000000
max    99.000000

Missing Values:
Date      0
Page      0
Visits    0
dtype: int64

import pandas as pd
import matplotlib.pyplot as plt
```

+ Code + Text

✓ 0s [6] 25% 25.000000
50% 47.000000
75% 72.000000
max 99.000000

Missing Values:
Date 0
Page 0
Visits 0
dtype: int64

RAM ✓
Disk

```
import pandas as pd
import matplotlib.pyplot as plt

# Load the dataset from the CSV file
df = pd.read_csv("website_traffic_dataset.csv")

# Group data by page and calculate the total visits for each page
page_visits = df.groupby("Page")["Visits"].sum().reset_index()

# Create a bar chart
plt.figure(figsize=(10, 6))
plt.bar(page_visits["Page"], page_visits["Visits"])
plt.xlabel("Page")
plt.ylabel("Visits")
plt.title("Website Traffic by Page")
plt.xticks(rotation=45) # Rotate x-axis labels for better readability
plt.show()
```

