INFOSYS SPRINGBOARD

kids' screentime patterns to uncover using data Visualization

Problem Statement:

Analyse kids' screentime patterns to uncover trends by age, gender, location type (urban/rural), device type, day-of-week, and activity category using data visualization. The goal is to present clear, actionable insights for parents, educators, and policymakers.

LOAD THE DATASET

Source: Kaggle — Indian Kids Screentime 2025

https://www.kaggle.com/datasets/ankushpanday2/indian-kids-screentime-2025

```
import pandas as pd
import numpy as np
from pathlib import Path


file_path = Path(r"D:\Infosys SpringBoard\data kaggle\Indian_Kids_Screen_Time.csv")
df = pd.read_csv(file_path)


print("Shape:", df.shape)
display(df.head())
display(df.dtypes)
```

```
Shape: (9712, 8)
```

	Age	Gender	Avg_Daily_Screen_Time_hr	Primary_Device	Exceeded_Recommended_Limit	Educational_t
0	14	Male	3.99	Smartphone	True	
1	11	Female	4.61	Laptop	True	
2	18	Female	3.73	TV	True	
3	15	Female	1.21	Laptop	False	
4	12	Female	5.89	Smartphone	True	

DATA CLEANING AND PREPROCESSING:

- Remove exact duplicate rows

```
df = df.drop_duplicates(keep='first')
print("After dropping exact duplicates, shape:", df.shape)
df.head()
```

```
After dropping exact duplicates, shape: (9668, 8)
```

	Age	Gender	Avg_Daily_Screen_Time_hr	Primary_Device	Exceeded_Recommended_Limit	Educational_t
0	14	Male	3.99	Smartphone	True	
1	11	Female	4.61	Laptop	True	
2	18	Female	3.73	TV	True	
3	15	Female	1.21	Laptop	False	
4	12	Female	5.89	Smartphone	True	

- Drop the rows having Avg_Daily_Screen_Time_hr=0

```
df = df[df["Avg_Daily_Screen_Time_hr"] != 0]

print("New shape:", df.shape)
display(df.head(20))
```

```
New shape: (9474, 8)
```

	Age	Gender	Avg_Daily_Screen_Time_hr	Primary_Device	Exceeded_Recommended_Limit	Educational_
0	14	male	3.99	smartphone	True	
1	11	female	4.61	laptop	True	
2	18	female	3.73	tv	True	
3	15	female	1.21	laptop	False	
4	12	female	5.89	smartphone	True	
5	14	female	4.88	smartphone	True	
6	17	male	2.97	tv	False	
7	10	male	2.74	tv	True	
8	14	male	4.61	laptop	True	
9	18	male	3.24	tablet	True	
10	18	male	3.53	tablet	True	

- Making all characters in lowercase of Gender column and Trim whitespace

```
str_cols = df.select_dtypes(include=['object']).columns.tolist()
for c in str_cols:
    df[c] = df[c].astype(str).str.strip().replace({'nan': np.nan})
    df[c] = df[c].where(df[c].isna(), df[c].str.lower())


if 'gender' in df.columns:
    gender_map = {
        'boy': 'male', 'm': 'male', 'male': 'male',
        'girl': 'female', 'f': 'female', 'female': 'female'
    }
    df['gender'] = df['gender'].map(gender_map).fillna(df['gender'])  # keep others unchanged
    print("Gender value counts after mapping:")
    display(df['gender'].value_counts(dropna=False))
df.head(20)
```

	Age	Gender	Avg_Daily_Screen_Time_hr	Primary_Device	Exceeded_Recommended_Limit	Educational_
0	14	male	3.99	smartphone	True	
1	11	female	4.61	laptop	True	
2	18	female	3.73	tv	True	
3	15	female	1.21	laptop	False	
4	12	female	5.89	smartphone	True	
5	14	female	4.88	smartphone	True	
6	17	male	2.97	tv	False	

- Deriving new fields from Educational_to_Recreational_Ratio

```
df["Recreational_Time_hr"] = df["Avg_Daily_Screen_Time_hr"] / (df["Educational_to_Recreational_Ratio"] + 1)
df["Educational_Time_hr"] = df["Avg_Daily_Screen_Time_hr"] - df["Recreational_Time_hr"]

df["Recreational_Time_hr"] = df["Recreational_Time_hr"].round(2)
df["Educational_Time_hr"] = df["Educational_Time_hr"].round(2)

df.head()
```

Primary_Device	Exceeded_Recommended_Limit	Educational_to_Recreational_Ratio	Health_Impacts	Urban_or_Rural	Recreational_Time_hr	Educational_Time_hr
smartphone	True	0.42	poor sleep, eye strain	urban	2.81	1.18
laptop	True	0.30	poor sleep	urban	3.55	1.06
tv	True	0.32	poor sleep	urban	2.83	0.90
laptop	False	0.39	NaN	urban	0.87	0.34
smartphone	True	0.49	poor sleep, anxiety	urban	3.95	1.94

- Changing datatype of Screen time into hh:mm form

```
def decimal_hours_to_hhmm(decimal_hours):
    if pd.isna(decimal_hours):
        return np.nan
    try:
        total_minutes = int(round(decimal_hours * 60))
        hours = total_minutes // 60
        minutes = total_minutes % 60
        return f"{hours:d}:{minutes:02d}"
    except Exception:
        return np.nan

if 'Avg_Daily_Screen_Time_hr' in df.columns:

    original_values = df['Avg_Daily_Screen_Time_hr'].copy()


    df['Avg_Daily_Screen_Time_hr'] = df['Avg_Daily_Screen_Time_hr'].apply(decimal_hours_to_hhmm)


    print("Screen time converted from decimal hours to hh:mm format:")
    display(df[['Avg_Daily_Screen_Time_hr']].head(10))
```

```
Screen time converted from decimal hours to hh:mm format:
```

	Avg_Daily_Screen_Time_hr
0	3:59
1	4:37
2	3:44
3	1:13
4	5:53
5	4:53
6	2:58
7	2:44

- Age Bands column is created for better visualization

```
age_bins = [0, 5, 8, 11, 14, 18]
age_labels = ['0-5', '6-8', '9-11', '12-14', '15-18']


df['Age_Band'] = pd.cut(df['Age'], bins=age_bins, labels=age_labels, right=True)
df.head()
```

evice	Exceeded_Recommended_Limit	Educational_to_Recreational_Ratio	Health_Impacts	Urban_or_Rural	Recreational_Time_hr	Educational_Time_hr	Age_Band
hone	True	0.42	poor sleep, eye strain	urban	2.81	1.18	12–14
ıptop	True	0.30	poor sleep	urban	3.55	1.06	9–11
tv	True	0.32	poor sleep	urban	2.83	0.90	15–18
ıptop	False	0.39	NaN	urban	0.87	0.34	15–18
hone	True	0.49	poor sleep, anxiety	urban	3.95	1.94	12–14

SAVE THE PRE-PROCESSED AND CLEANED DATA

```
print("Final shape:", df.shape)
print("Final missing counts:")
display(df.isna().sum().sort_values(ascending=False).head(30))

#  Save cleaned dataset
clean_path = file_path.parent / (file_path.stem + "_week2.csv")
df.to_csv(clean_path, index=False)
print("Cleaned file saved to:", clean_path)
```

```
Final shape: (9474, 11)
Final missing counts:
Health_Impacts                      2986
Gender                                 0
Age                                    0
Avg_Daily_Screen_Time_hr               0
Primary_Device                         0
Exceeded_Recommended_Limit             0
Educational_to_Recreational_Ratio      0
Urban_or_Rural                         0
Recreational_Time_hr                   0
Educational_Time_hr                    0
Age_Band                               0
dtype: int64
Cleaned file saved to: D:\Infosys SpringBoard\data kaggle\Indian_Kids_Screen_Time_week2.csv
```

SUMMARY

The dataset, sourced from Kaggle, undergoes preprocessing: removing duplicate rows, excluding entries with zero average daily screen time, and standardizing the gender column by trimming whitespace and converting to lowercase. Additional transformations include deriving new fields from the Educational-to-Recreational Ratio, converting screen time into hh:mm format, and creating age bands for clearer visual representation. These steps ensure the data is clean, consistent, and ready for insightful analysis. The final output is a refined dataset that supports meaningful visualizations and interpretations for children’s digital habits. This structured approach empowers stakeholders to make informed decisions about screen exposure and its educational or recreational balance.