ScreenSense:Kids' Screentime Visualization
WEEK 1 & WEEK 2

1. Importing Libraries

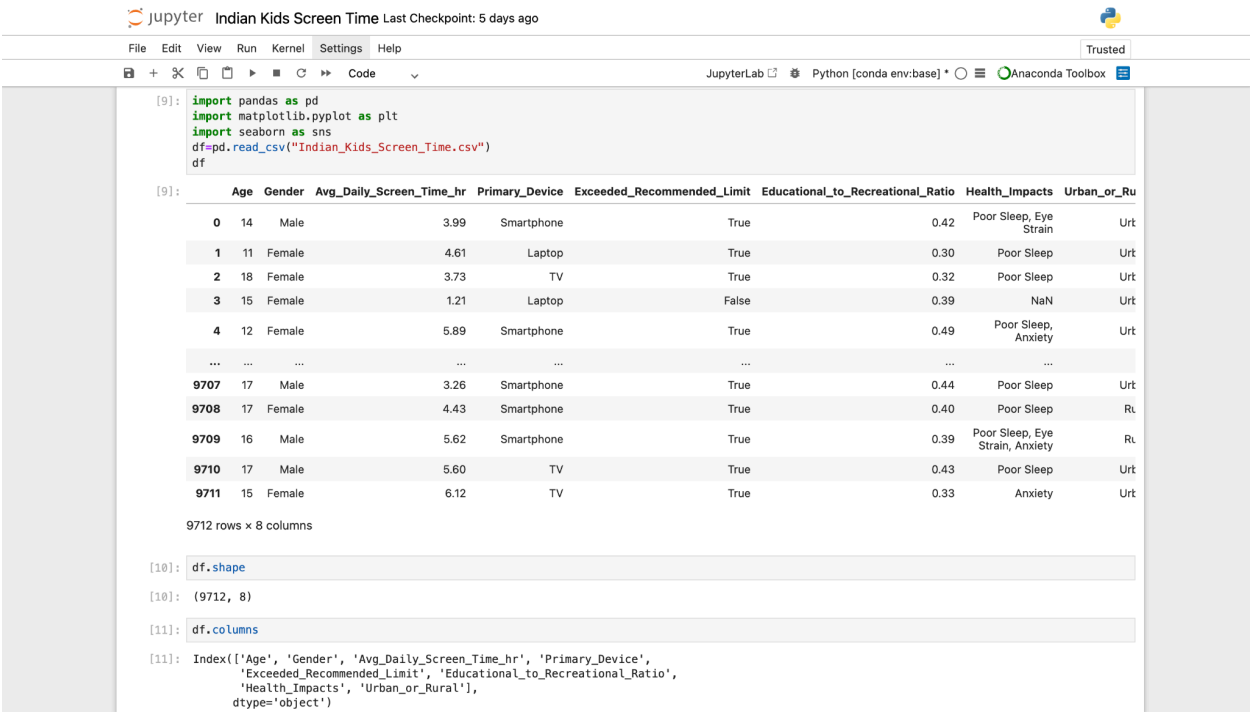
This section imports essential Python libraries:

- `pandas` for data manipulation,
- `matplotlib.pyplot` and `seaborn` for visualization,
- `statistics` for calculating mode, and
- `numpy` for numerical operations.

2. Loading the Dataset

The dataset `Indian_Kids_Screen_Time.csv` is read into a pandas DataFrame using `pd.read_csv()`.

We print its shape, column names, and unique values in the *Health_Impacts* column to understand its structure.



jupyter Indian Kids Screen Time Last Checkpoint: 5 days ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python [conda env:base] * Anaconda Toolbox

[9]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
df=pd.read_csv("Indian_Kids_Screen_Time.csv")
df
```

[9]:

	Age	Gender	Avg_Daily_Screen_Time_hr	Primary_Device	Exceeded_Recommended_Limit	Educational_to_Recreational_Ratio	Health_Impacts	Urban_or_Ru
0	14	Male	3.99	Smartphone	True	0.42	Poor Sleep, Eye Strain	Urt
1	11	Female	4.61	Laptop	True	0.30	Poor Sleep	Urt
2	18	Female	3.73	TV	True	0.32	Poor Sleep	Urt
3	15	Female	1.21	Laptop	False	0.39	NaN	Urt
4	12	Female	5.89	Smartphone	True	0.49	Poor Sleep, Anxiety	Urt
...	...	...	...	...	...	...	...	
9707	17	Male	3.26	Smartphone	True	0.44	Poor Sleep	Urt
9708	17	Female	4.43	Smartphone	True	0.40	Poor Sleep	Ru
9709	16	Male	5.62	Smartphone	True	0.39	Poor Sleep, Eye Strain, Anxiety	Ru
9710	17	Male	5.60	TV	True	0.43	Poor Sleep	Urt
9711	15	Female	6.12	TV	True	0.33	Anxiety	Urt

9712 rows × 8 columns

[10]:
```
df.shape
```

[10]:
```
(9712, 8)
```

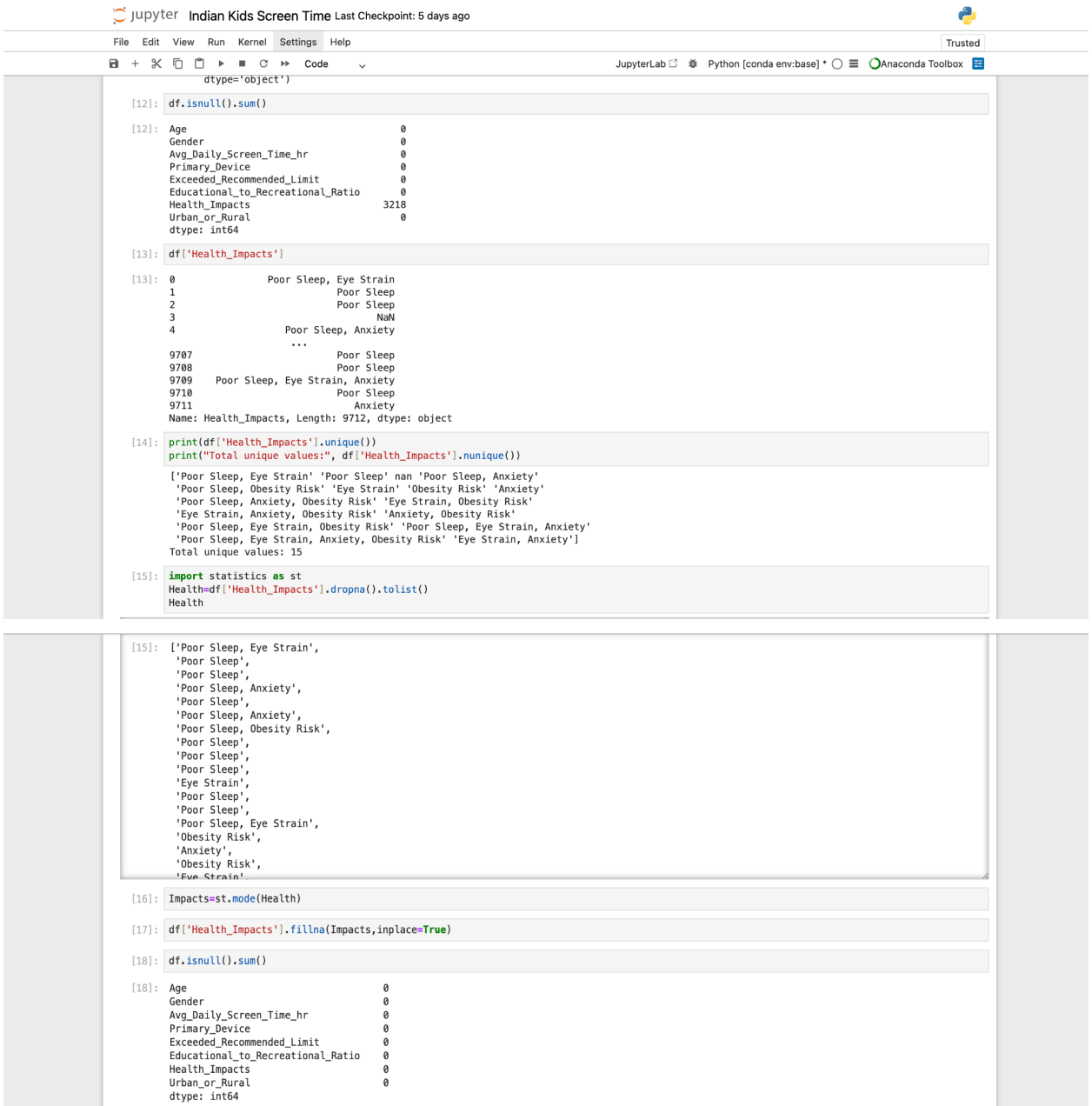
[11]:
```
df.columns
```

[11]:
```
Index(['Age', 'Gender', 'Avg_Daily_Screen_Time_hr', 'Primary_Device',
       'Exceeded_Recommended_Limit', 'Educational_to_Recreational_Ratio',
       'Health_Impacts', 'Urban_or_Rural'],
      dtype='object')
```

3. Handling Missing Values in ‘Health_Impacts’

Missing values are handled by replacing them with the most frequent (mode) value in the column.

This ensures no data is lost and the distribution remains balanced.



```
jupyter Indian Kids Screen Time Last Checkpoint: 5 days ago
File Edit View Run Kernel Settings Help                                   Trusted
Code                                   JupyterLab  Python [conda env:base] *  Anaconda Toolbox
          dtype='object')

[12]: df.isnull().sum()

[12]: Age                                    0
      Gender                                 0
      Avg_Daily_Screen_Time_hr               0
      Primary_Device                         0
      Exceeded_Recommended_Limit             0
      Educational_to_Recreational_Ratio      0
      Health_Impacts                      3218
      Urban_or_Rural                         0
      dtype: int64

[13]: df['Health_Impacts']

[13]: 0               Poor Sleep, Eye Strain
      1                           Poor Sleep
      2                           Poor Sleep
      3                                  NaN
      4                  Poor Sleep, Anxiety
                              ...
      9707                        Poor Sleep
      9708                        Poor Sleep
      9709    Poor Sleep, Eye Strain, Anxiety
      9710                        Poor Sleep
      9711                           Anxiety
      Name: Health_Impacts, Length: 9712, dtype: object

[14]: print(df['Health_Impacts'].unique())
      print("Total unique values:", df['Health_Impacts'].nunique())

      ['Poor Sleep, Eye Strain' 'Poor Sleep' nan 'Poor Sleep, Anxiety'
       'Poor Sleep, Obesity Risk' 'Eye Strain' 'Obesity Risk' 'Anxiety'
       'Poor Sleep, Anxiety, Obesity Risk' 'Eye Strain, Obesity Risk'
       'Eye Strain, Anxiety, Obesity Risk' 'Anxiety, Obesity Risk'
       'Poor Sleep, Eye Strain, Obesity Risk' 'Poor Sleep, Eye Strain, Anxiety'
       'Poor Sleep, Eye Strain, Anxiety, Obesity Risk' 'Eye Strain, Anxiety']
      Total unique values: 15

[15]: import statistics as st
      Health=df['Health_Impacts'].dropna().tolist()
      Health

[15]: ['Poor Sleep, Eye Strain',
       'Poor Sleep',
       'Poor Sleep',
       'Poor Sleep, Anxiety',
       'Poor Sleep',
       'Poor Sleep, Anxiety',
       'Poor Sleep, Obesity Risk',
       'Poor Sleep',
       'Poor Sleep',
       'Poor Sleep',
       'Eye Strain',
       'Poor Sleep',
       'Poor Sleep',
       'Poor Sleep, Eye Strain',
       'Obesity Risk',
       'Anxiety',
       'Obesity Risk',
       'Eye Strain',

[16]: Impacts=st.mode(Health)

[17]: df['Health_Impacts'].fillna(Impacts,inplace=True)

[18]: df.isnull().sum()

[18]: Age                                    0
      Gender                                 0
      Avg_Daily_Screen_Time_hr               0
      Primary_Device                         0
      Exceeded_Recommended_Limit             0
      Educational_to_Recreational_Ratio      0
      Health_Impacts                         0
      Urban_or_Rural                         0
      dtype: int64
```

4. Descriptive Statistics

`df.describe()` generates a statistical summary (mean, std, min, max, etc.) of numerical columns, helping in understanding the data range and variability.

```
[19]: df.describe()
```

[19]:

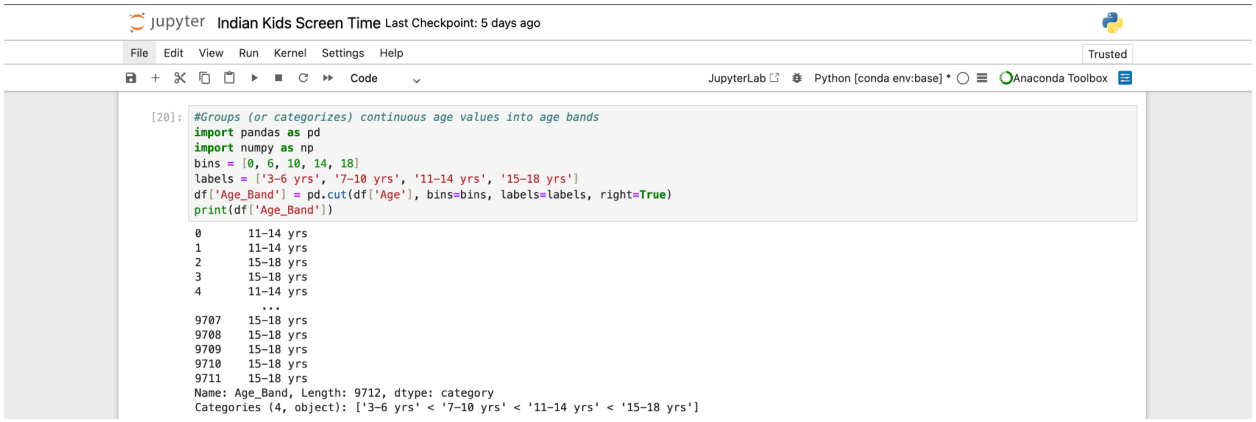
	Age	Avg_Daily_Screen_Time_hr	Educational_to_Recreational_Ratio
count	9712.000000	9712.000000	9712.000000
mean	12.979201	4.352837	0.427226
std	3.162437	1.718232	0.073221
min	8.000000	0.000000	0.300000
25%	10.000000	3.410000	0.370000
50%	13.000000	4.440000	0.430000
75%	16.000000	5.380000	0.480000
max	18.000000	13.890000	0.600000

5. Creating Age Bands

Continuous age values are categorized into 4 bands using `pd.cut()`:

- 3–6 yrs
- 7–10 yrs
- 11–14 yrs
- 15–18 yrs

This helps analyze screen time patterns across different age groups.

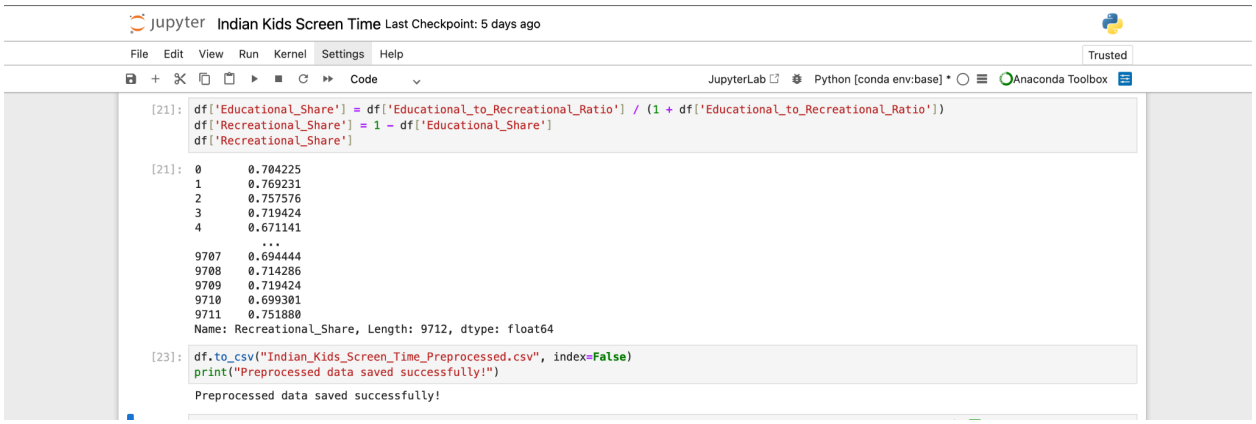


6. Computing Educational and Recreational Share

The ratio `Educational_to_Recreational_Ratio` is converted into percentage shares:

- `Educational_Share` = Educational proportion

- `Recreational_Share` = Remaining portion (1 - Educational_Share)
  This gives a clearer comparison between educational and recreational screen usage.



7. Saving the Preprocessed Data

The cleaned and feature-enriched dataset is saved as `Indian_Kids_Screen_Time_Preprocessed.csv` for further analysis or visualization.