FDSL / Assignment2_partA.ipynb

Fizashaikh63 Add files via upload 87e1bc7 · 5 minutes ago

906 lines (906 loc) · 23.8 KB

Preview Code Blame Raw

In [12]:

```
#PANDAS Library
import pandas as pd

# Creating a Series
s = pd.Series([12, 13, 14, 15, 15], index=['a','b','c','d','e'])

print("Pandas Series:")
print(s)
print("\nAccess single value:", s['c'])
```

```
Pandas Series:
a    12
b    13
c    14
d    15
e    15
dtype: int64

Access single value: 14
```

In [36]:

```
type(s)
```

Out[36]:

```
pandas.core.series.Series
```

In [27]:

```
# Creating a DataFrame
data = {
    'Name': ['Fiza', 'Humera', 'vaishnavi','hindavi','ruchita','Anam'],
    'Age': [18, 19, 20,18,19,20],
    'Marks': [88, 92, 85,89,90,95]
}

df = pd.DataFrame(data)

print("Pandas DataFrame:")
print(df)
print("\nAccess single column (as Series):")
print(df['Name'])
```

```
Pandas DataFrame:
        Name  Age  Marks
0       Fiza   18     88
1     Humera   19     92
2  vaishnavi   20     85
3    hindavi   18     89
4    ruchita   19     90
5       Anam   20     95

Access single column (as Series):
0         Fiza
1       Humera
2    vaishnavi
3      hindavi
4      ruchita
5         Anam
Name: Name, dtype: object
```

In [28]:

```
type(df)
```

Out[28]:

```
pandas.core.frame.DataFrame
```

In [29]:

```
#perform arithmatic operation
s = pd.Series([11, 23, 31, 45, 52])
print("Original:\n", s)
print("\nMultiplied by 2:\n", s * 2)
print("\n Divide by 2:\n",s/2)
print("\n mode by 2:\n",s%2)
print("\n Substraction by 5\n",s-7)
```

```
Original:
 0    11
1    23
2    31
3    45
4    52
dtype: int64

Multiplied by 2:
 0     22
1     46
2     62
3     90
4    104
dtype: int64

 Divide by 2:
 0     5.5
1    11.5
2    15.5
3    22.5
4    26.0
dtype: float64

 mode by 2:
 0    1
1    1
2    1
3    1
4    0
dtype: int64

 Substraction by 5
 0     4
1    16
2    24
3    38
4    45
dtype: int64
```

In [46]:

```
print(df['Name'] ) # select single column
```

```
0         Fiza
1       Humera
2    vaishnavi
3      hindavi
4      ruchita
5         Anam
Name: Name, dtype: object
```

In [42]:

```
print(df[['Name','Marks']])   # select multiple columns
```

```
        Name  Marks
0       Fiza     88
1     Humera     92
2  vaishnavi     85
3    hindavi     89
4    ruchita     90
5       Anam     95
```

In [43]:

```
print(df.iloc[0] )       # first row by index
```

```
Name      Fiza
Age         18
Marks       88
Name: 0, dtype: object
```

In [44]:

```
print(df.iloc[1:3])          # rows by index range
```

```
        Name  Age  Marks
1     Humera   19     92
2  vaishnavi   20     85
```

In [45]:

```
print(df.loc[df['Marks']>70]) # filter condition
```

```
        Name  Age  Marks
0       Fiza   18     88
1     Humera   19     92
2  vaishnavi   20     85
3    hindavi   18     89
4    ruchita   19     90
5       Anam   20     95
```

In [31]:

```
df.describe()          # summary statistics (mean, std, min, max, etc.)
df['Marks'].mean()     # average marks
df['Marks'].median()   # median
df['Marks'].max()      # maximum value
df['Marks'].min()      # minimum value
df['Marks'].sum()      # sum of column
df['Marks'].value_counts()  # frequency count
```

Out[31]:

```
Marks
88    1
92    1
85    1
89    1
90    1
95    1
Name: count, dtype: int64
```

In [49]:

```
data=pd.read_csv("Indian_Kids_Screen_Time.csv")
```

In [50]:

```
data.to_csv("updated.csv", index=False)
```

In [52]:

```
data.head()
```

Out[52]:

	Age	Gender	Avg_Daily_Screen_Time_hr	Primary_Device	Exceeded_Recommended_
0	14	Male	3.99	Smartphone	

1	11	Female	4.61	Laptop
2	18	Female	3.73	TV
3	15	Female	1.21	Laptop
4	12	Female	5.89	Smartphone

In [53]:

```
data.tail()
```

Out[53]:

	Age	Gender	Avg_Daily_Screen_Time_hr	Primary_Device	Exceeded_Recommend
9707	17	Male	3.26	Smartphone	
9708	17	Female	4.43	Smartphone	
9709	16	Male	5.62	Smartphone	
9710	17	Male	5.60	TV	
9711	15	Female	6.12	TV	

In [19]:

```
print(data.shape)
```

```
(9712, 8)
```

In [54]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9712 entries, 0 to 9711
Data columns (total 8 columns):
 #   Column                             Non-Null Count  Dtype  
---  ------                             --------------  -----  
 0   Age                                9712 non-null   int64  
 1   Gender                             9712 non-null   object 
 2   Avg_Daily_Screen_Time_hr           9712 non-null   float64
 3   Primary_Device                     9712 non-null   object 
 4   Exceeded_Recommended_Limit         9712 non-null   bool   
 5   Educational_to_Recreational_Ratio  9712 non-null   float64
 6   Health_Impacts                     6494 non-null   object 
 7   Urban_or_Rural                     9712 non-null   object 
dtypes: bool(1), float64(2), int64(1), object(4)
memory usage: 540.7+ KB
```

In [55]:

```
data.describe()
```

Out[55]:

	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

In [24]:

```
print(data.dtypes)
```

```
Age                                    int64
Gender                                object
Avg_Daily_Screen_Time_hr             float64
Primary_Device                        object
Exceeded_Recommended_Limit              bool
Educational_to_Recreational_Ratio    float64
Health_Impacts                        object
Urban_or_Rural                        object
dtype: object
```

In [25]:

```
print(data.columns)
```

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

In [26]:

```
print(data.index)
```

```
RangeIndex(start=0, stop=9712, step=1)
```

In [56]:

```
data.describe()          # summary statistics (mean, std, min, max, etc.)
data['Age'].mean()     # average marks
data['Age'].median()   # median
data['Age'].max()      # maximum value
data['Age'].min()      # minimum value
data['Age'].sum()      # sum of column
data['Age'].value_counts()  # frequency count
```

Out[56]:

```
Age
17    919
8     912
13    910
14    896
9     885
10    877
16    876
12    867
11    866
15    864
18    840
Name: count, dtype: int64
```