Python Program on Data and Variables.

October 24, 2024

[10]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

[11]:
```
df = pd.read_csv("advertising.csv")
df
```

[11]:

	Daily Time Spent on Site	Age	Area Income	Daily Internet Usage	\
0	68.95	35	61833.90	256.09	
1	80.23	31	68441.85	193.77	
2	69.47	26	59785.94	236.50	
3	74.15	29	54806.18	245.89	
4	68.37	35	73889.99	225.58	
..	…	…	…	…	
995	72.97	30	71384.57	208.58	
996	51.30	45	67782.17	134.42	
997	51.63	51	42415.72	120.37	
998	55.55	19	41920.79	187.95	
999	45.01	26	29875.80	178.35	

	Ad Topic Line	City	Male	\
0	Cloned 5thgeneration orchestration	Wrightburgh	0	
1	Monitored national standardization	West Jodi	1	
2	Organic bottom-line service-desk	Davidton	0	
3	Triple-buffered reciprocal time-frame	West Terrifurt	1	
4	Robust logistical utilization	South Manuel	0	
..	…	…	…	
995	Fundamental modular algorithm	Duffystad	1	
996	Grass-roots cohesive monitoring	New Darlene	1	
997	Expanded intangible solution	South Jessica	1	
998	Proactive bandwidth-monitored policy	West Steven	0	
999	Virtual 5thgeneration emulation	Ronniemouth	0	

	Country	Timestamp	Clicked on Ad
0	Tunisia	2016-03-27 00:53:11	0
1	Nauru	2016-04-04 01:39:02	0
2	San Marino	2016-03-13 20:35:42	0

```
3                      Italy  2016-01-10 02:31:19              0
4                    Iceland  2016-06-03 03:36:18              0
..                       …                  …             …
995                  Lebanon  2016-02-11 21:49:00              1
996  Bosnia and Herzegovina  2016-04-22 02:07:01              1
997                 Mongolia  2016-02-01 17:24:57              1
998                Guatemala  2016-03-24 02:35:54              0
999                   Brazil  2016-06-03 21:43:21              1

[1000 rows x 10 columns]
```

[12]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 10 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Daily Time Spent on Site  1000 non-null   float64
 1   Age                       1000 non-null   int64
 2   Area Income               1000 non-null   float64
 3   Daily Internet Usage      1000 non-null   float64
 4   Ad Topic Line             1000 non-null   object
 5   City                      1000 non-null   object
 6   Male                      1000 non-null   int64
 7   Country                   1000 non-null   object
 8   Timestamp                 1000 non-null   object
 9   Clicked on Ad             1000 non-null   int64
dtypes: float64(3), int64(3), object(4)
memory usage: 78.2+ KB
```

[14]:
```
print("\nBasic Statistics:")
print(df.describe())
```

```
Basic Statistics:
       Daily Time Spent on Site          Age   Area Income  \
count               1000.000000  1000.000000   1000.000000
mean                  65.000200    36.009000  55000.000080
std                   15.853615     8.785562  13414.634022
min                   32.600000    19.000000  13996.500000
25%                   51.360000    29.000000  47031.802500
50%                   68.215000    35.000000  57012.300000
75%                   78.547500    42.000000  65470.635000
max                   91.430000    61.000000  79484.800000

       Daily Internet Usage         Male  Clicked on Ad
count           1000.000000  1000.000000     1000.00000
```

```
mean             180.000100     0.481000        0.50000
std               43.902339     0.499889        0.50025
min              104.780000     0.000000        0.00000
25%              138.830000     0.000000        0.00000
50%              183.130000     0.000000        0.50000
75%              218.792500     1.000000        1.00000
max              269.960000     1.000000        1.00000
```

[13]:
```
df=df.drop(["Ad Topic Line","City","Country","Timestamp"],axis=1)
print("\nMedian of each column:")
print(df.median())
```

```
Median of each column:
Daily Time Spent on Site       68.215
Age                            35.000
Area Income                 57012.300
Daily Internet Usage          183.130
Male                            0.000
Clicked on Ad                   0.500
dtype: float64
```

[15]:
```
print("\nStandard Deviation of each column:")
print(df.std())
```

```
Standard Deviation of each column:
Daily Time Spent on Site       15.853615
Age                             8.785562
Area Income                 13414.634022
Daily Internet Usage           43.902339
Male                            0.499889
Clicked on Ad                   0.500250
dtype: float64
```

[16]:
```
print("\nMissing values in each column:")
print(df.isnull().sum())
```

```
Missing values in each column:
Daily Time Spent on Site    0
Age                         0
Area Income                 0
Daily Internet Usage        0
Male                        0
Clicked on Ad               0
dtype: int64
```

[17]:
```
df.isnull().sum()
```

[17]:
```
Daily Time Spent on Site    0
Age                         0
Area Income                 0
Daily Internet Usage        0
Male                        0
Clicked on Ad               0
dtype: int64
```

[18]:
```
df_filled = df.fillna(df.mean())
df_filled
```

[18]:
```
     Daily Time Spent on Site  Age  Area Income  Daily Internet Usage  Male  \
0                       68.95   35     61833.90                256.09     0
1                       80.23   31     68441.85                193.77     1
2                       69.47   26     59785.94                236.50     0
3                       74.15   29     54806.18                245.89     1
4                       68.37   35     73889.99                225.58     0
..                        ...  ...          ...                   ...   ...
995                     72.97   30     71384.57                208.58     1
996                     51.30   45     67782.17                134.42     1
997                     51.63   51     42415.72                120.37     1
998                     55.55   19     41920.79                187.95     0
999                     45.01   26     29875.80                178.35     0

     Clicked on Ad
0                0
1                0
2                0
3                0
4                0
..             ...
995              1
996              1
997              1
998              0
999              1

[1000 rows x 6 columns]
```

[22]:
```
# Grouping by 'Male' and calculating mean, sum, and count for other numeric␣
  ↪columns
grouped = df.groupby('Male').agg({
    'Daily Time Spent on Site': ['mean', 'sum', 'count'],
    'Area Income': ['mean', 'sum', 'count'],
    'Daily Internet Usage': ['mean', 'sum', 'count']
})
```

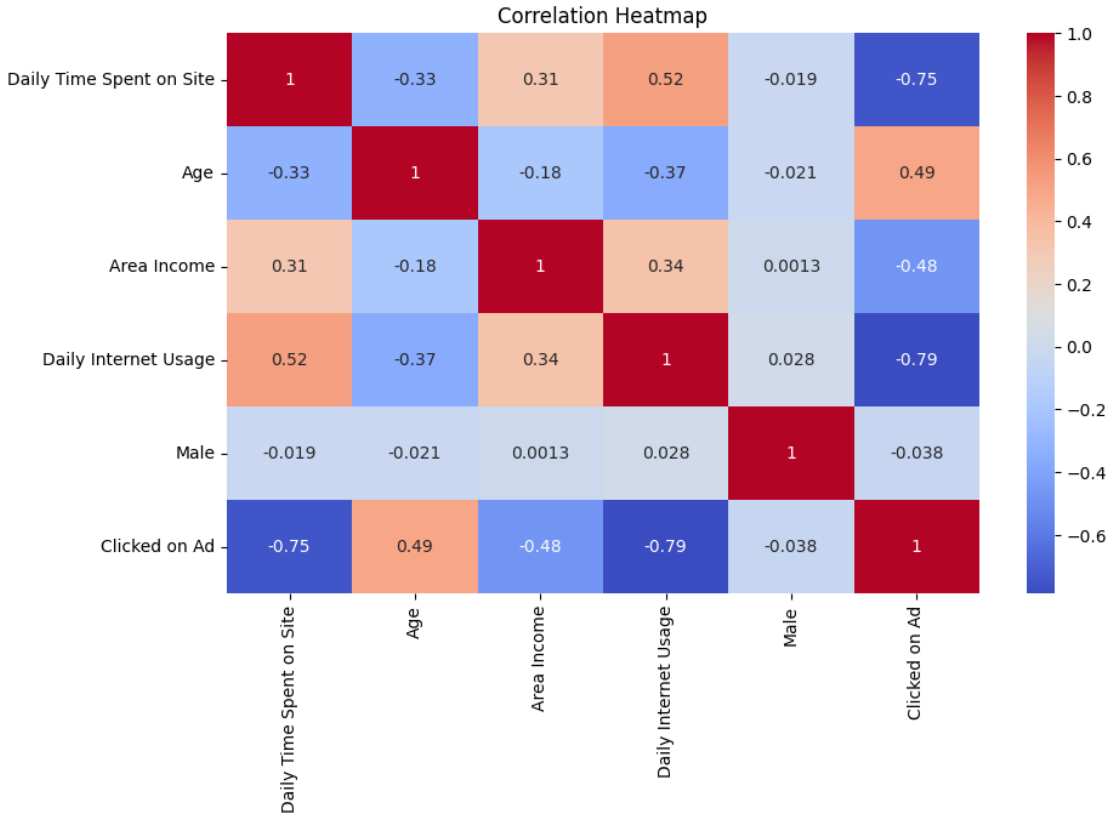
```
print("\nGrouped Data Statistics by 'Male':")
print(grouped)
```

```
Grouped Data Statistics by 'Male':
     Daily Time Spent on Site                    Area Income               \
                         mean       sum count          mean           sum
Male
0                   65.289287  33885.14   519  54982.931407  28536141.40
1                   64.688274  31115.06   481  55018.417214  26463858.68

           Daily Internet Usage
     count                 mean      sum count
Male
0      519           178.816763  92805.9   519
1      481           181.276923  87194.2   481
```

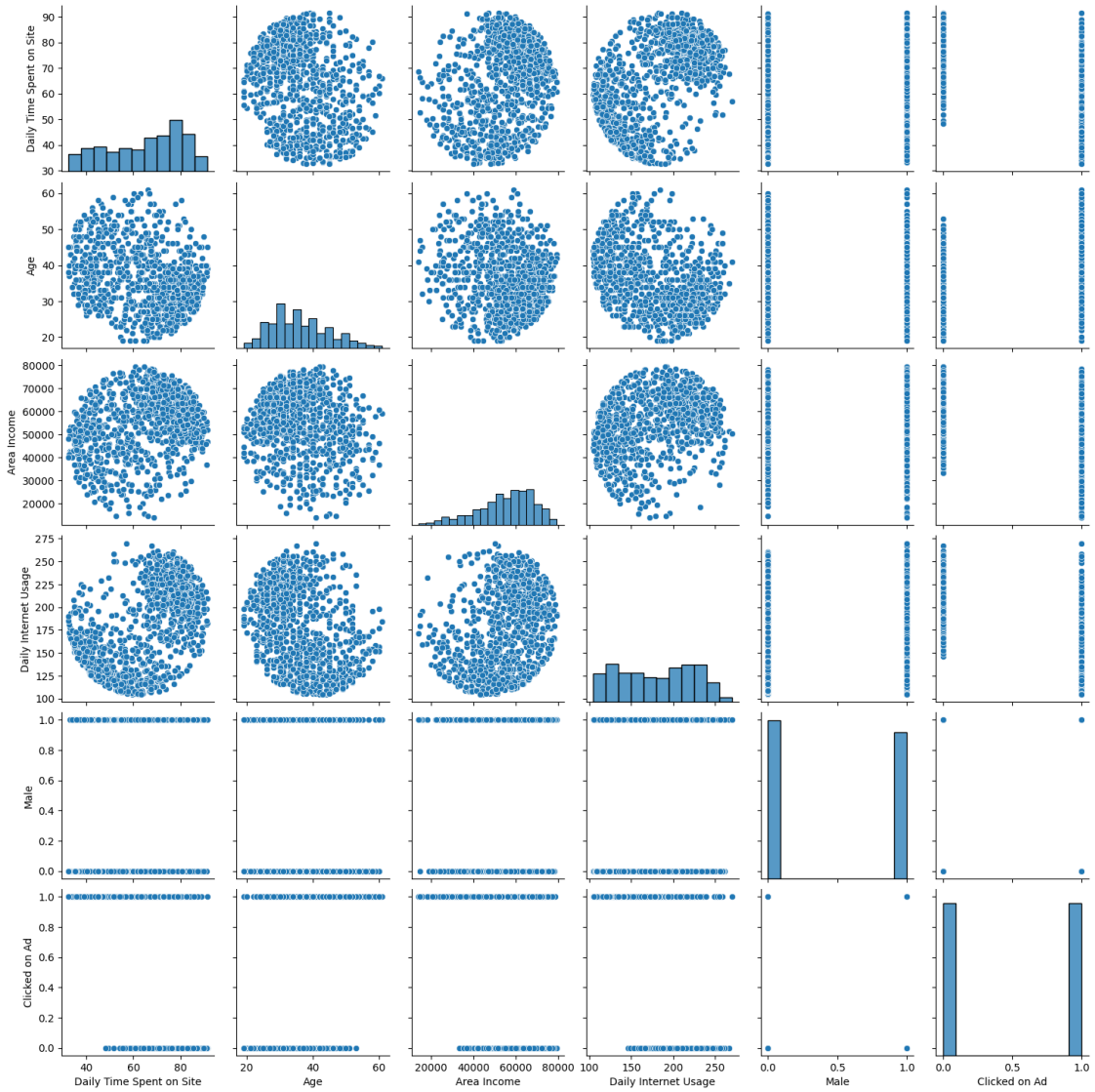
[23]:
```
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()
df_normalized = pd.DataFrame(scaler.fit_transform(df_filled), columns=df_filled.
 ↪columns)

print("\nNormalized Data:")
print(df_normalized.head())
```

```
Normalized Data:
   Daily Time Spent on Site       Age  Area Income  Daily Internet Usage  \
0                  0.617882  0.380952     0.730472              0.916031
1                  0.809621  0.285714     0.831375              0.538746
2                  0.626721  0.166667     0.699200              0.797433
3                  0.706272  0.238095     0.623160              0.854280
4                  0.608023  0.380952     0.914568              0.731323

   Male  Clicked on Ad
0   0.0            0.0
1   1.0            0.0
2   0.0            0.0
3   1.0            0.0
4   0.0            0.0
```

[24]:
```
# Correlation heatmap
plt.figure(figsize=(10, 6))
sns.heatmap(df.corr(), annot=True, cmap="coolwarm")
plt.title('Correlation Heatmap')
plt.show()
```



[25]:
```
# Pair plot
sns.pairplot(df)
plt.show()
```


Daily Time Spent on Site
Age
Area Income
Daily Internet Usage
Male
Clicked on Ad

[ ]: