ai-program-day-3

May 5, 2024

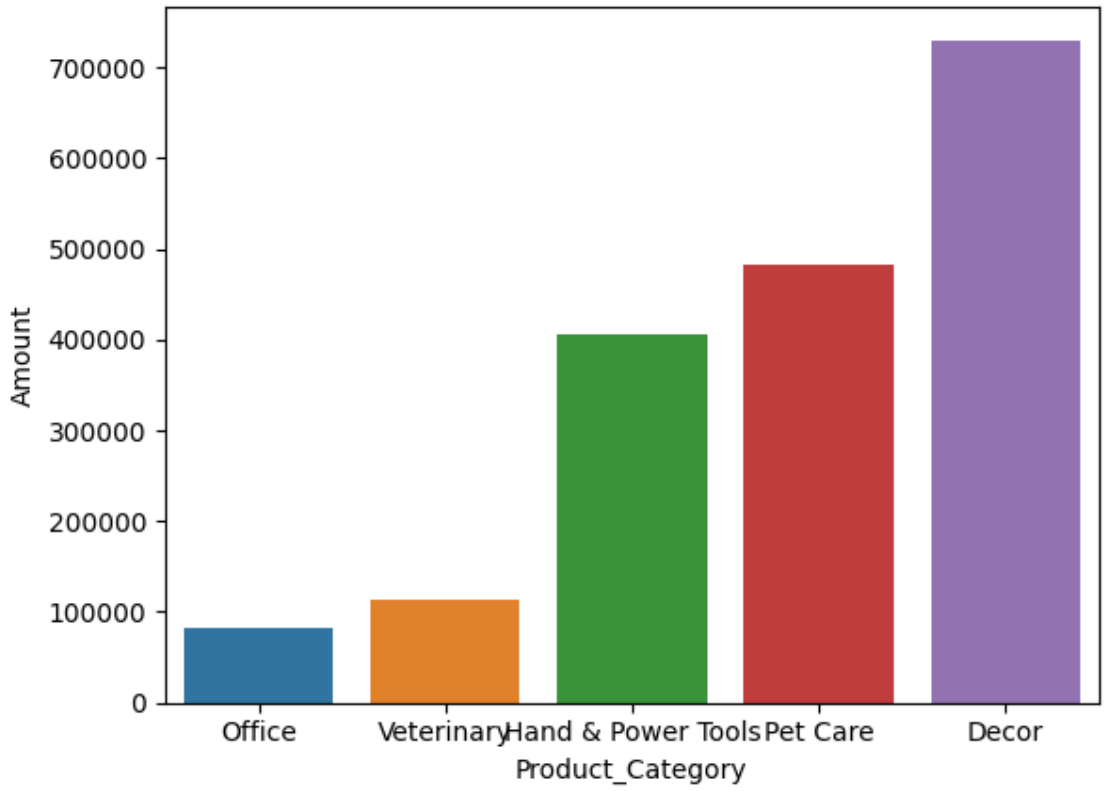
[46]:
```
#find total sales amount by product category
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
df=pd.read_csv("Amazon sales Data.csv",encoding='unicode_escape')
sales_prod=df.groupby(["Product_Category"], as_index=False)["Amount"].sum()
sales_prod
```

[46]:

	Product_Category	Amount
0	Auto	1958609.99
1	Beauty	1959484.00
2	Books	1061478.00
3	Clothing & Apparel	16495019.00
4	Decor	730360.00
5	Electronics & Gadgets	15643846.00
6	Food	33933883.50
7	Footwear & Shoes	15575209.45
8	Furniture	5440051.99
9	Games & Toys	4331694.00
10	Hand & Power Tools	405618.00
11	Household items	1569337.00
12	Office	81936.00
13	Pet Care	482277.00
14	Sports Products	3635933.00
15	Stationery	1676051.50
16	Tupperware	1155642.00
17	Veterinary	112702.00

[47]:
```
import seaborn as sns
sales_prod=df.groupby(["Product_Category"], as_index=False)["Amount"].sum().
 ↪sort_values(ascending=True,by="Amount").head()
sales_prod
sns.barplot(x="Product_Category",y="Amount",data=sales_prod)
```

[47]:
```
<Axes: xlabel='Product_Category', ylabel='Amount'>
```



[48]:
```
# write a function that will give sum of numbers in the list.
#without using any predefined python function
mylist=[11,12,33,24,55,68,71,82,29]
print("The give list is:")
print(mylist)
list_length=len(mylist)
sumOfElements=0
for i in range(list_length):
    sumOfElements=sumOfElements+mylist[i]
print("Sum of all elements in the list is:",sumOfElements)
```

```
The give list is:
[11, 12, 33, 24, 55, 68, 71, 82, 29]
Sum of all elements in the list is: 385
```

[49]:
```
def list_sum(x):
    sum=0
    for i in x:
        sum=sum+i
    return sum
print(list_sum([4,2,3]))
```

9

[50]:
```
#write the function that will give count of vowels present in the given string/
 ↪text.
#not to use any predefined python libraries/functions
def count_vowels(x):
    c=0
    v=['a','e','i','o','u']
    for i in x:
        if i in v:
            c+=1
    return c
count_vowels("jwegfweihdsakjffi")
```

[50]: 5

[51]:
```
import pandas as pd
```

[52]:
```
df_bookings=pd.read_csv("fact_bookings.csv")
```

[53]:
```
df_bookings.head()
```

[53]:

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platform	ratings_given	booking_status	revenue_generated	revenue_realized
0	May012216558RT11	16558	27-04-22	1/5/2022	2/5/2022	-3.0	RT1	direct online	1.0	Checked Out	10010	10010
1	May012216558RT12	16558	30-04-22	1/5/2022	2/5/2022	2.0	RT1	others	NaN	Cancelled	9100	3640
2	May012216558RT13	16558	28-04-22	1/5/2022	4/5/2022	2.0	RT1	logtrip	5.0	Checked Out	9100000	9100
3	May012216558RT14	16558	28-04-22	1/5/2022	2/5/2022	-2.0	RT1	others	NaN	Cancelled	9100	3640
4	May012216558RT15	16558	27-04-22	1/5/2022	2/5/2022	4.0	RT1	direct online	5.0	Checked Out	10920	10920

[54]:
```
df_bookings=pd.read_csv("dim_date.csv")
df_bookings.head()
```

[54]:
```
         date  mmm yy week no  day_type
0  01-May-22  May 22    W 19   weekend
1  02-May-22  May 22    W 19  weekeday
2  03-May-22  May 22    W 19  weekeday
3  04-May-22  May 22    W 19  weekeday
4  05-May-22  May 22    W 19  weekeday
```

[55]:
```
df_bookings=pd.read_csv("dim_hotels.csv")
df_bookings.head()
```

[55]:
```
   property_id  property_name  category    city
0        16558   Atliq Grands    Luxury   Delhi
1        16559  Atliq Exotica    Luxury  Mumbai
2        16560     Atliq City  Business   Delhi
3        16561      Atliq Blu    Luxury   Delhi
4        16562      Atliq Bay    Luxury   Delhi
```

[56]:
```
df_bookings=pd.read_csv("dim_rooms.csv")
df_bookings.head()
```

[56]:
```
  room_id    room_class
0     RT1      Standard
1     RT2         Elite
2     RT3       Premium
3     RT4  Presidential
```

[57]:
```
df_bookings=pd.read_csv("fact_aggregated_bookings.csv")
df_bookings.head()
```

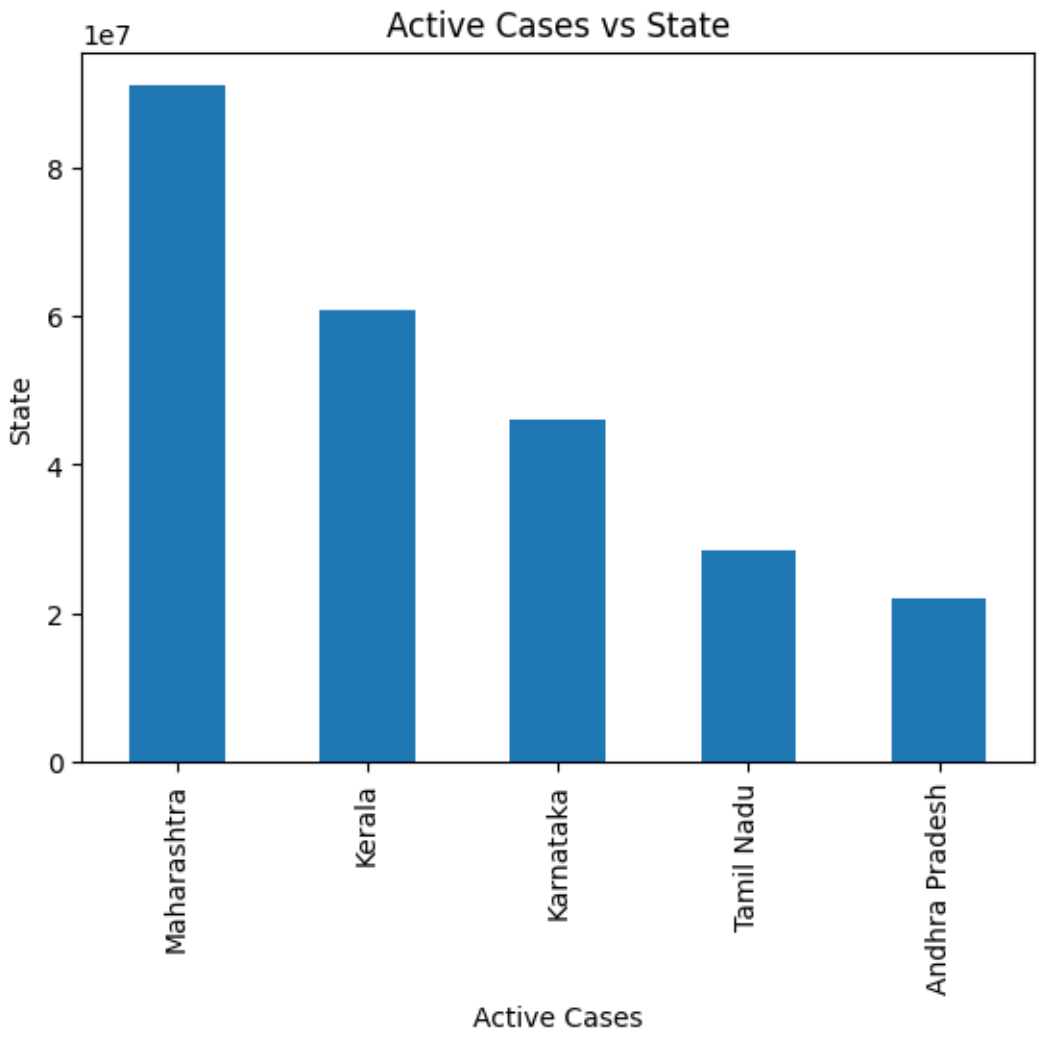
[57]:
```
   property_id check_in_date room_category  successful_bookings  capacity
0        16559      1-May-22           RT1                   25      30.0
1        19562      1-May-22           RT1                   28      30.0
2        19563      1-May-22           RT1                   23      30.0
3        17558      1-May-22           RT1                   30      19.0
4        16558      1-May-22           RT1                   18      19.0
```

[58]:
```
df_bookings=pd.read_csv("new_data_august.csv")
df_bookings.head()
```

[58]:
```
   property_id  property_name  category       city room_category room_class  \
0        16559  Atliq Exotica    Luxury     Mumbai           RT1   Standard
1        19562      Atliq Bay    Luxury  Bangalore           RT1   Standard
2        19563   Atliq Palace  Business  Bangalore           RT1   Standard
3        19558   Atliq Grands    Luxury  Bangalore           RT1   Standard
4        19560     Atliq City  Business  Bangalore           RT1   Standard

  check_in_date  mmm yy week no  day_type  successful_bookings  capacity  \
```

```
0     01-Aug-22  Aug-22    W 32  weekeday                    30         30
1     01-Aug-22  Aug-22    W 32  weekeday                    21         30
2     01-Aug-22  Aug-22    W 32  weekeday                    23         30
3     01-Aug-22  Aug-22    W 32  weekeday                    30         40
4     01-Aug-22  Aug-22    W 32  weekeday                    20         26

     occ%
0  100.00
1   70.00
2   76.67
3   75.00
4   76.92
```

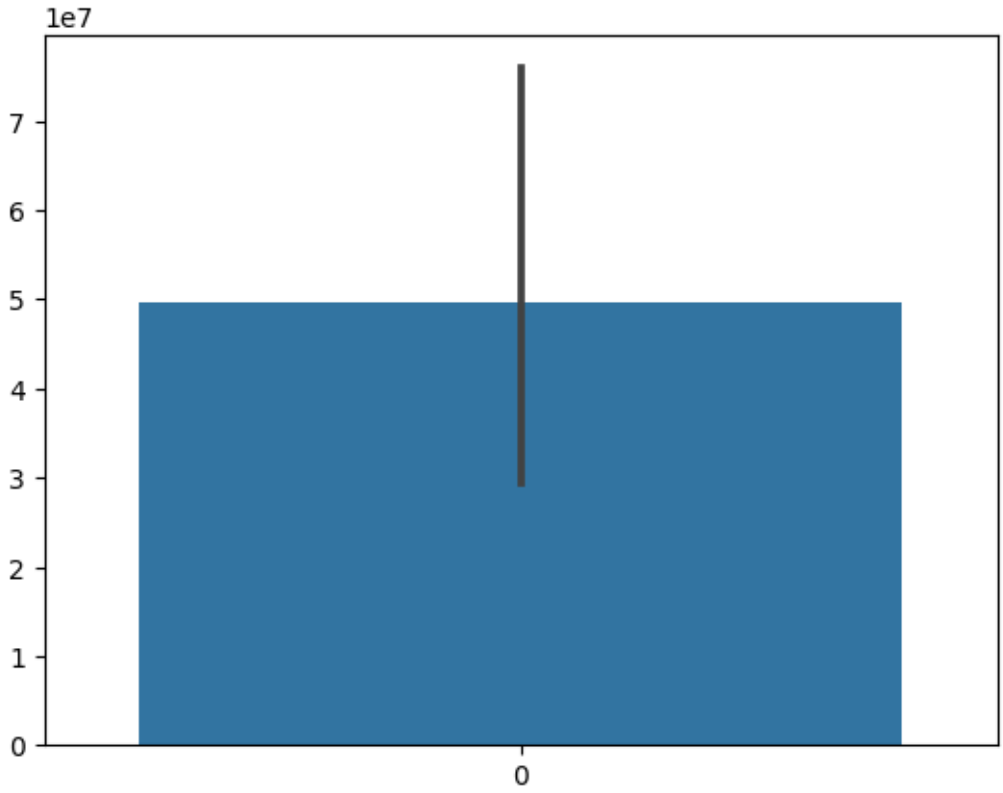
[5]:
```
import pandas as pd
import matplotlib.pyplot as plt
df=pd.read_excel('Covid_Data.xlsx')
df_new=df.groupby('State')['Active Cases'].sum()
df_new=df_new.sort_values(ascending=False).head(5)
df_new.plot(kind="bar")
plt.xlabel("Active Cases")
plt.ylabel("State")
plt.title("Active Cases vs State")
plt.show()
```



[7]:
```
#Active cases in states
import pandas as pd
import seaborn as sns
df_new=df.groupby('State')['Active Cases'].sum()
df_new=df_new.sort_values(ascending=False).head(5)
sns.barplot(data=df_new)
```

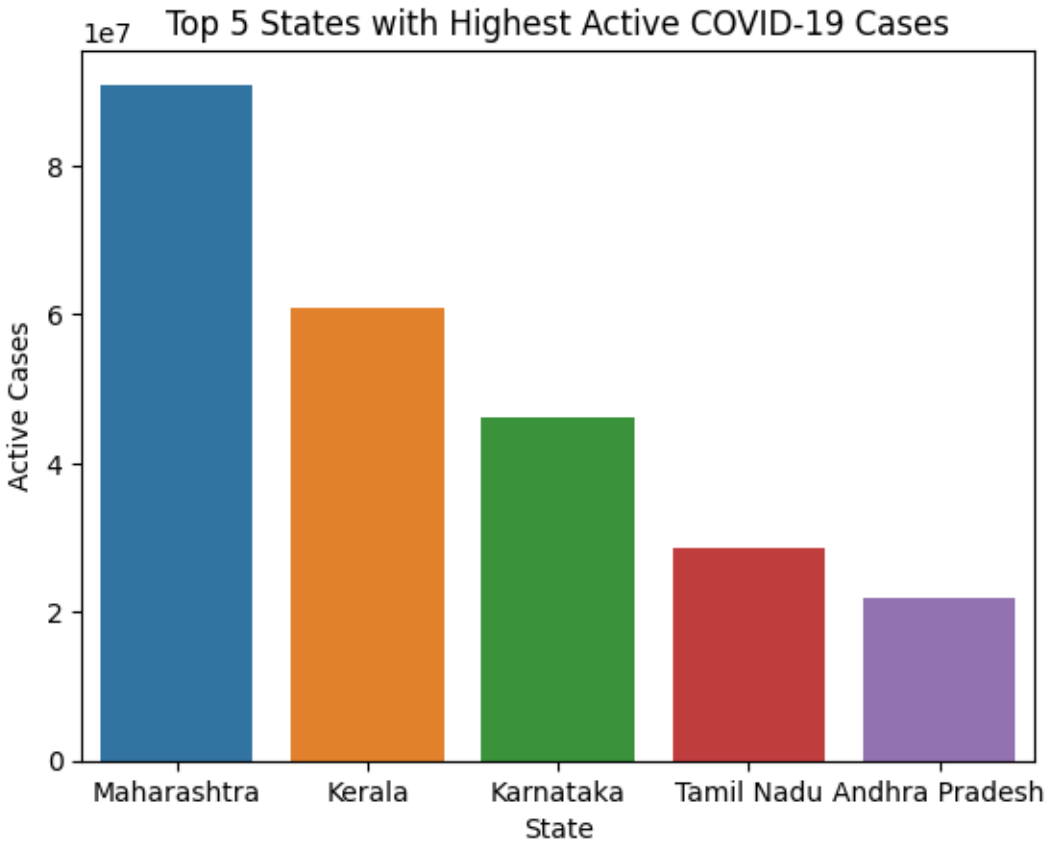
```
C:\ProgramData\anaconda3\Lib\site-packages\seaborn\categorical.py:486:
FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a
future version, integer keys will always be treated as labels (consistent with
DataFrame behavior). To access a value by position, use `ser.iloc[pos]`
  if np.isscalar(data[0]):
```

[7]: <Axes: >



[10]:
```
import pandas as pd
import seaborn as sns

# Read the data from Excel file
df = pd.read_excel('Covid_Data.xlsx')

# Group by state and sum up active cases
df_new = df.groupby('State')['Active Cases'].sum()

# Select the top 5 states with the highest number of active cases
df_new = df_new.sort_values(ascending=False).head(5).reset_index()

# Plotting the bar chart using Seaborn
sns.barplot(x='State', y='Active Cases', data=df_new)

# Adding labels and title
plt.xlabel("State")
plt.ylabel("Active Cases")
plt.title("Top 5 States with Highest Active COVID-19 Cases")

# Show the plot
```

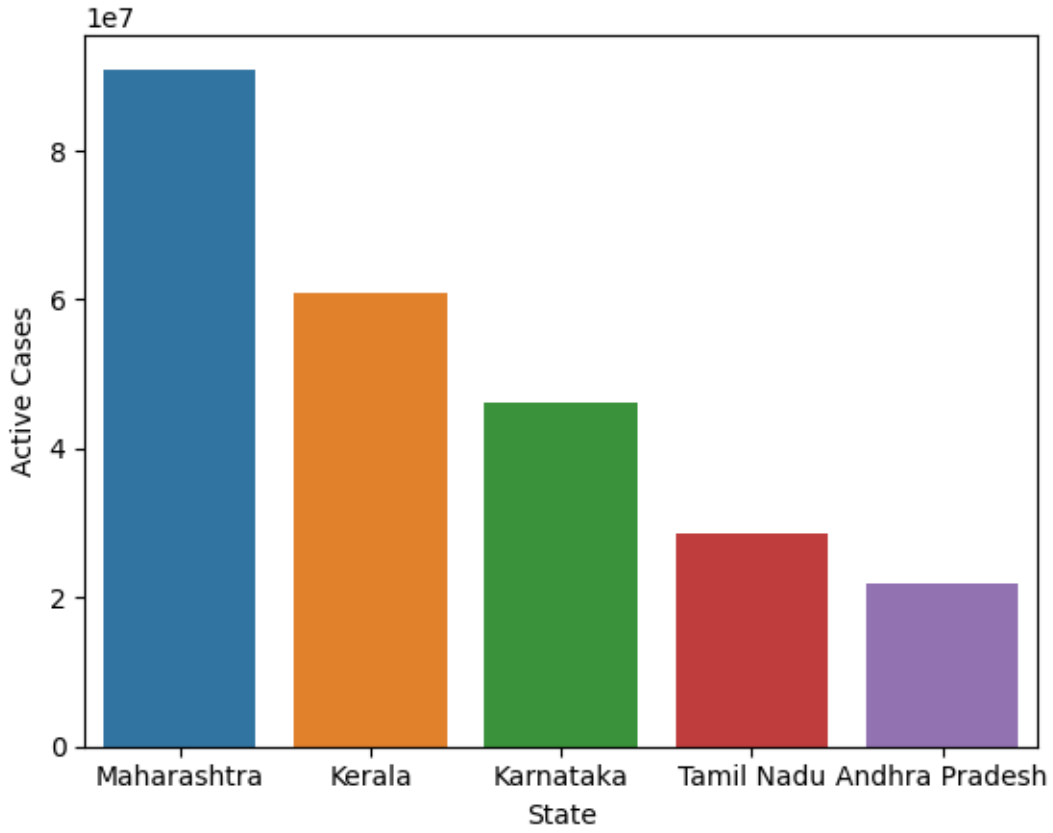
```
plt.show()
```



[11]:
```
import seaborn as sns
df=pd.read_excel('Covid_Data.xlsx')
Covid_Data=df.groupby(['State'],as_index=False)['Active Cases'].sum().
  ↪sort_values(ascending=False,by="Active Cases").head()
print(Covid_Data.head())
sns.barplot(x="State",y="Active Cases",data=Covid_Data)
```

```
            State  Active Cases
20     Maharashtra      90912115
16          Kerala      60837834
15       Karnataka      46067883
31      Tamil Nadu      28627946
1   Andhra Pradesh      22001587
```

[11]: <Axes: xlabel='State', ylabel='Active Cases'>



[64]:
```
import pandas as pd
import random
data={"Name":
 ↪["Sathwika","Suresh","Vaishnavi","Tejashwini","Deepak","Rajashree","Yashwanth","Anil"],"Gen
 ↪["F","M","F","F","M","F","M","M"],"Age":[23,24,20,21,24,22,18,22],
                "Height(in cm)":[156,165,185,152,153,165,162,176],"Weight":
 ↪[45,68,55,52,62,50,51,60]}
df=pd.DataFrame(data)
for _ in range(5):
    name=''.join(random.choice('abcdefghijklmnopqrstuvwxyz') for _ in range(5))
    gender=random.choice(["M","F"])
    age=random.randint(18,30)
    height=random.randint(150,200)
    weight=random.randint(40,100)
print(df)
```

```
         Name Gender  Age  Height(in cm)  Weight
0    Sathwika      F   23            156      45
1      Suresh      M   24            165      68
2   Vaishnavi      F   20            185      55
```

```
3  Tejashwini      F   21            152      52
4      Deepak      M   24            153      62
5   Rajashree      F   22            165      50
6   Yashwanth      M   18            162      51
7        Anil      M   22            176      60
```

[66]:
```
df['BMI']=df['Weight']/((df['Height(in cm)']/100)**2)
average_bmi=df['BMI'].mean()
print(df)
print("\nAverage BMI: {:.2f}".format(average_bmi))
```

```
         Name Gender  Age  Height(in cm)  Weight        BMI
0    Sathwika      F   23            156      45  18.491124
1      Suresh      M   24            165      68  24.977043
2   Vaishnavi      F   20            185      55  16.070124
3  Tejashwini      F   21            152      52  22.506925
4      Deepak      M   24            153      62  26.485540
5   Rajashree      F   22            165      50  18.365473
6   Yashwanth      M   18            162      51  19.433013
7        Anil      M   22            176      60  19.369835

Average BMI: 20.71
```

[ ]: