**Day 1:**

#Import the csv library

import csv

#Open the CSV file

with open('C:\\Users\\MVSR\\Desktop\\Dataset.csv','r')as csv\_file:

###Use the csv.reader object to read the CSV file.

csv\_reader=csv.reader(csv\_file)

### Extract the field names

header = next(csv\_reader)

# Extract the field names

#Load the data from a CSV file into memory using the CSV module.

#Open the CSV file

with open('C:\\Users\\MVSR\\Desktop\\Dataset.csv','r',encoding='utf8')as csv\_file:

#Use the csv.reader object to read the CSV file.

csv\_reader=csv.reader(csv\_file)

#Retrieve a name of listing, host\_name, description, host\_location, and the date the host was created for an individual host by host\_id

#To retrieve the data by host Id

id = input("Enter host ID")

#gives us al list of everything in col 0

for row in csv\_reader:

if row[0] == id:

print("Name of listing:"," ",row[1])

print("Host Name:"," ",row[3])

print("Discription:"," ",row[2])

print("Host Location:"," ",row[5])

print("Date the host was created:"," ",row[4])

break

**DAY 2:**

#Load the data from a CSV file into memory using the CSV module.

#Open the CSV file

with open('C:\\Users\\MVSR\\Desktop\\Dataset.csv','r',encoding='utf8')as csv\_file:

#Use the csv.reader object to read the CSV file.

csv\_reader=csv.reader(csv\_file)

###Retrieve host\_name, property\_type, price, minimum\_nights, and maximum\_nights of all Airbnb listing for a specified location

###To retrieve the data by specified location

location = input("Enter the location")

###Gives us a list of location in col 5

count=0

for row in csv\_reader:

if row[5] == location:

count+=1;

print("Details ",count," for same location for ")

print("Host Name:"," ",row[3])

print("Property type"," ",row[13])

print("Price:"," ",row[20])

print("Minimum nights:"," ",row[21])

print("Maximum nights:"," ",row[22])

###Retrieve room\_type, accommodates, bathrooms, bedroom, and beds of all Airbnb listing for a specified property type

with open('C:\\Users\\MVSR\\Desktop\\Dataset.csv','r',encoding='utf8')as csv\_file:

###Use the csv.reader object to read the CSV file.

csv\_reader=csv.reader(csv\_file)

value=0

property = input("Enter property type: ")

count=0

for row in csv\_reader:

if row[13] == property:

count=count+1;

print("Details ",count," for same property type ")

print("Room type:"," ",row[14])

print("Accommodates:"," ",row[15])

print("Bathrooms:"," ",row[16])

print("Bedroom:"," ",row[17])

print("Beds:"," ",row[18])

###Retrieve room\_type, accommodates, bathrooms, bedroom, and beds of all Airbnb listing for a specified property type

with open('C:\\Users\\MVSR\\Desktop\\Dataset.csv','r',encoding='utf8')as csv\_file:

###Use the csv.reader object to read the CSV file.

csv\_reader=csv.reader(csv\_file)

#Retrieve specific columns of your choice related to an individual host by location

host = input("Enter host name : ")

count=0

for row in csv\_reader:

if row[3] == host:

count=count+1;

print("Details ",count," for same host name : ")

print("Name:"," ",row[1])

print("Host Response:"," ",row[6])

print("Room Type:"," ",row[15])

**DAY 3:**

"""Load the data from a CSV file into memory using the pandas module. The

path to the file will be specified by the user then use these loaded data to

perform following tasks"""

import pandas as pd

df = pd.read\_csv('C:\\Users\\MVSR\\Desktop\\Dataset.csv')

print("loaded")

###Identifying the most popular amenities or features that Airbnb guests are looking for

### The pandas value\_counts() function is used to get the count of each unique value in a pandas series. You can use it to get the counts and then extract the value with the most counts using idxmax() function.

print("The most popular aminities that Airbnb guests are looking are :")

df['amenities'].value\_counts().idxmax()

**Day 4:**

###Analyse the average price of stay in each location

#### Calculating mean on groupby

avgprice = df.groupby('host\_location')['price'].mean()

###To print all the values

print("The average price of stay in each location are:")

print(avgprice.to\_string())

###analse the average review score for each location

avgreview = df.groupby('host\_location')['review\_scores\_rating'].mean()

print("the average review score for each location are:")

print(avgreview.to\_string())

#Analyse to get insightful information based on your own selection (should not be the same as the previous requirements)

avgbedrooms = df.groupby('host\_location')['bedrooms'].mean()

print("the average bedrooms for each location are:")

print(avgbedrooms.to\_string())

**DAY 5:**

import matplotlib.pyplot as plt

import pandas as pd

df = pd.read\_csv('C:\\Users\\MVSR\\Desktop\\Dataset.csv')

###Display the proportion of number of bedrooms of Airbnb listing using pie chart

###let’s first sort the indices and counts for our bedroom column:

label=df.bedrooms.value\_counts().index

count=df.bedrooms.value\_counts().values

###TO enlarge our chart to view it clearly figsize parameter in the figure() function to set the dimensions of the figure in inches.

###plt.figure(1, figsize=(20,10))

##pie chart

plt.pie(count,labels=label)

###Use plt.title() for setting a plot title

###Use plt.legend() for the observation variables

###Use plt.show() for displaying the plot

plt.title('Proportion of bedrooms')

plt.legend()

plt.show()

**DAY 6:**

###Display the number of listings for each room type using bar chart

df['room\_type'].value\_counts().plot(kind='bar')

###Display the relationship between accommodates and price using scatter plot

#import seaborn as sns

subplot=df[['accommodates','price']]

scatter=subplot.plot(x ='accommodates', y = 'price', kind='scatter', c='green');

scatter.set\_title('Relationship between accommodates and price');

scatter.figure.savefig('Relationship.png')

#sns.scatterplot(x="accommodates", y="price", data=df);

**DAY 7:**

###Display Airbnb prices from 2019 - 2022 with line chart using subplots (one year per plot)

# Make a list of columns

import datetime

#import seaborn as sns

# convert the host\_since Date to datetime

df["host\_since"]= pd.to\_datetime(df["host\_since"])

## add a column for Year to store only 'year'

df['Year'] = df['host\_since'].dt.strftime('%Y')

df['Year']

#extract the year from 2019 to 2022 and strore in year1

Year1=pd.DataFrame(df.loc[(df['Year']>='2019')&(df['Year']<='2022')])

Year1.columns

plt.plot(Year1['Year']=='2019',Year1['price'],color='g',label='2019')

plt.xlabel("year 2019")

plt.ylabel("price")

plt.legend()

plt.show()

**DAY 8:**

fig,ax=plt.subplots(1,4,sharey=True)

plt.subplot(141)

plt.plot(Year1['Year']=='2019',Year1['price'],color='g',label='2019')

plt.xlabel("year 2019")

plt.subplot(142)

plt.plot(Year1['Year']=='2020',Year1['price'],color='g',label='2020')

plt.xlabel("year 2020")

plt.subplot(143)

plt.plot(Year1['Year']=='2021',Year1['price'],color='g',label='2021')

plt.xlabel("year 2021")

plt.subplot(144)

plt.plot(Year1['Year']=='2022',Year1['price'],color='g',label='2022')

plt.xlabel("year 2022")

plt.ylabel("price")

plt.legend()

plt.show()

**DAY 9:**

#Year1['bedrooms'].plot(kind="hist",bins=5)

fig.ax=plt.subplots(figsize=(5,5))

plt.hist(Year1['bedrooms'],bins=5)