

```
In [1]: # Import Python Libraries as per requirement
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
# Import Healthcare dataset using pandas
data = pd.read_csv(r"C:\Users\ADMIN\Downloads\healthcare_dataset.csv",encoding="latin1")
df=pd.DataFrame(data)
# Check if there is any null values in the datasets
df.isnull().sum() # There is no null value in the dataset
# Make a data visualizations using seaborn and matplotlib libraries
# Group by insurance provider and their client percentage
insurance_provider_client=df.groupby("Insurance Provider")["Name"].count()
# Make a pie charts by insurance provider and their client perecentage
plt.pie(insurance_provider_client,labels=insurance_provider_client.index,autopct="%1.1f%%",startangle=120,colors=["skyblue","lightgreen","red","orange","yellow"])
plt.title("Insurance Provider By Clients Counts")
plt.show()
# top 10 Group by Hospitals and thier Number of clients
hospitals_clients=df.groupby("Hospital")["Name"].count()
top_10_hospitals_clients=hospitals_clients.sort_values(ascending=False).head(10)
# Make a Bar Chart on top 10 Hospitals Basend on Number Of Clients
plt.figure(figsize=(5,8))
top_10_hospitals_clients.plot(kind="bar",color="blue")
# Add Labels and title
plt.title("Top 10 Hospitals Based on Number of Clients")
plt.xlabel("Hospitals Names")
plt.ylabel("Number Of Clients")
plt.show()
# Group by insurance provider and avg Bill amount
avg_insurance_provider_bill_amount=df.groupby("Insurance Provider")["Billing Amount"].mean()
# Make a Line chart
plt.figure(figsize=(8,10))
plt.plot(avg_insurance_provider_bill_amount.index,avg_insurance_provider_bill_amount.values,marker="o",linestyle="-",color="black")
# Add Labels and title
plt.title("Insurance Provider Average Bill Amount")
plt.xlabel("Insurance Provider")
plt.ylabel("Avg Bill Amount")
plt.show()
# Group by Gender and Admission Type and count the number of clients
Gender_admission_type_clients_count=df.groupby(["Gender","Admission Type"])[ "Name"].count()
# Create the bar plot
Gender_admission_type_clients_count.unstack().plot(kind="bar", figsize=(10, 6), colormap="viridis")
# Add Labels and title
plt.xlabel("Gender and admission type")
plt.ylabel("Number of Clients")
plt.title("Number of Clients by Gender and Admission Type")
# Show the plot
plt.show()
# Group by Gender Medical Condition Based on Clients counts
Gender_medical_conditon_counts=df.groupby(["Gender","Medical Condition"])[ "Name"].count()
# Create the heatmap chart
heatmap_data = Gender_medical_conditon_counts.unstack() # we want to reshape a series into dataframe where rows correspond to genders and columns correspond to medcial condition
# Create the heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(heatmap_data, annot=True, fmt="d", cmap="coolwarm", linewidths=.5)
# Add Labels and title
plt.xlabel("Medical Condition")
plt.ylabel("Gender")
plt.title("Heatmap of Gender vs Medical Condition Counts")
# Show the plot
plt.show()
# Group by Gender Blood type based on number of clients
gender_blood_type_counts=df.groupby(["Gender","Blood Type"])[ "Name"].count()
gender_blood_type_counts.unstack().plot(kind="bar",figsize=(10,6),colormap="viridis")
# Add Labels and title
plt.title("Barplot of Gender vs Blood type counts")
plt.xlabel("Gender And Blood Type")
plt.ylabel("Number of Clients")
plt.show()
```



