```python
#!/usr/bin/env python
# coding: utf-8

# In[1]:

# importing packages
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt



# In[2]:


df = pd.read_csv('County_Mental_Health_Profiles__2006-2016.csv')
print(df)


# In[3]:


df.dtypes


# In[4]:


#incase types are different when importing from excel
##df = df.astype({'Recipient_Count_By_County':'int', 'Count Of  Recipients By F

##df = df.astype({'Paid_Claim_Total':'int'})

# In[5]:


#Cleaning the DATA
del df['Row_Created_Date_Time']

# In[6]:


# Dropping rows that do not have age group as either Adult or Child
#df[(df['Age_Group'] != 'ADULT') & (df['Age_Group'] != 'CHILD')]
df.drop(df[(df['Age_Group'] != 'ADULT') & (df['Age_Group'] != 'CHILD')].index,


# In[7]:


#Dropping rows that have 0 money claimed and negative units were served
#df[df['Units Total'] < 0]
#df[df['Paid Claim Total'] < 0]
df.drop(df[df['Units_Total'] < 0].index, inplace = True)
df.drop(df[df['Paid_Claim_Total'] < 0].index, inplace = True)
```

```
# In[8]:


print(df)

##Saving the modified file
##df.to_csv("Cleaned_dataset.csv")

# In[9]:


# Removing duplicates when the type of treatment is not considered
by_county = df[['County_Label','Age_Group','Recipient_Count_By_County']]
by_county=by_county.drop_duplicates()
by_county


# In[10]:


# Summarizing data based on county and number of patients in each county
county = by_county.groupby('County_Label', as_index = False).agg({'Recipient_Co
county = county.reset_index(drop = True)
top_counties=county.head(10)
plt.figure()
rg1=top_counties.plot.scatter(x='County_Label',y='Recipient_Count_By_County',s=
plt.xticks(rotation = 90)
plt.xlabel('County')
plt.ylabel('Number of patients')
plt.title('Top counties with most number of patients')
plt.show()



# In[11]:


# Calculating the total number patients in each age group
by_age = df.groupby('Age_Group', as_index =False).agg({'Recipient_Count_By_Cour
by_age


# In[12]:


# Summarizing the number of patients in each age group in each county
by_age_pc = by_county.groupby(['County_Label', 'Age_Group']).sum()
by_age_pc.head(10)



# In[13]:
```

```
#Calculating the number of patients that are treated by each type of treatment
by_treatment = df.groupby('Rate_Code_Group').agg({'Count_Of_Recipients_By_Rate_

by_treatment


plt.figure()
rg3 = by_treatment.plot.barh(legend=None)
plt.ylabel('Treatment Types')
plt.xlabel('Number of patients treated in Millions')
plt.title('Most commonly given treatment to the patients')

plt.show()

# # Visulizations

# In[14]:


# Top 5 counties with the most number of patients
top_county = county['County_Label'].tolist()
top_county = top_county[0:5]
print(top_county)


# In[39]:


# Calculating the number of patients in top 5 counties(based on number of patie
county_by_year = df[['Service_Year', 'County_Label', 'Age_Group','Recipient_Cou
county_by_year=county_by_year.drop_duplicates()
county_by_year = county_by_year[county_by_year['County_Label'].isin(top_county)
county_by_year = county_by_year.groupby(['Service_Year', 'County_Label'], as_in
county_by_year = county_by_year.pivot_table(index = ['Service_Year'], columns =
county_by_year = county_by_year.rename_axis(None, axis = 0)
county_by_year = county_by_year.rename_axis('Service_Year', axis = 1)
county_by_year


# In[49]:

plt.figure()
g1 = county_by_year.plot.barh(title='Number of patients in top 5 counties in ea
plt.xlabel('Number of patients')
plt.ylabel('Years')
plt.show()


# In[69]:


# Calculating the avg expenditure of each county on mental health issues
avg_cost_county = df[['County_Label', 'Paid_Claim_Total']]
avg_cost_county = avg_cost_county.groupby('County_Label', as_index = False).agg
```

```
avg_cost_county


# In[81]:


g2 = plt.plot(avg_cost_county['County_Label'], avg_cost_county['Paid_Claim_Tota

fig = plt.figure()
plt.bar(avg_cost_county['County_Label'], avg_cost_county['Paid_Claim_Total'])
plt.xticks(rotation = 90)
plt.show()

# In[18]:


# Calcualting the number of patients in each age group in each year
year_age = df[['Service_Year', 'County_Label', 'Age_Group','Recipient_Count_By_
year_age = year_age.drop_duplicates()
year_age = year_age.groupby(['Service_Year', 'Age_Group']).agg({'Recipient_Coun
year_age = year_age.pivot_table(index = ['Service_Year'], columns = 'Age_Group'
year_age = year_age.rename_axis(None, axis = 0)
year_age = year_age.rename_axis('Service_Year', axis = 1)
year_age


# In[66]:

plt.figure()
g3 = year_age.plot.bar(title = 'Number of patients in each age group in each ye
plt.xlabel('Years')
plt.ylabel('Number of patients')
plt.show()

# In[19]:


#Total expense on mental health by each county
county_expense = df[['Service_Year', 'County_Label', 'Paid_Claim_Total']]
county_expense = county_expense.groupby('County_Label').agg({'Paid_Claim_Total'
county_expense.head(10)


# In[20]:


# Most costliest Treatment based on expenditure only
treatment_cost = df[['Service_Year', 'Rate_Code_Group', 'Paid_Claim_Total']]
treatment_cost = treatment_cost.groupby('Rate_Code_Group').agg({'Paid_Claim_Tot
treatment_cost


# In[83]:

plt.figure()
```

```
treatment_cost.plot.barh()
plt.xlabel('Amount spent in Billion Dollars')
plt.ylabel('Type of Treatment')
plt.show()

# In[21]:


#Most costliest Treatment based on average expenditure for each treatment
treatment_avgcost = df[['Service_Year', 'Rate_Code_Group', 'Count_Of_Recipients
treatment_avgcost = treatment_avgcost.groupby('Rate_Code_Group').agg({'Count_Of
treatment_avgcost['Avg_Cost'] = treatment_avgcost.apply(lambda x: round(x['Paid
treatment_avgcost = treatment_avgcost.sort_values(by = 'Avg_Cost', ascending =
treatment_avgcost




# In[61]:


# Total number of patients that are treated each year
patients_year = df[['Service_Year','County_Label', 'Age_Group', 'Recipient_Coun
patients_year = patients_year.drop_duplicates()
patients_year = patients_year.groupby('Service_Year', as_index = False).agg({'R
patients_year


# In[64]:

plt.figure()

g6 = plt.plot(patients_year['Service_Year'], patients_year['Recipient_Count_By_
plt.title('Total number of patients that are treated in each year')
plt.xlabel('Service_Year')
plt.ylabel('Number of Patients Treated')
plt.show()
```