

PROJECT INTRODUCTION

The main objective of this project is to develop strong data analysis skills that enable us to handle complex data, which is crucial in today's technologies such as Data Science and Artificial Intelligence. Through this project, I have acquired various data analytic skills, including data cleaning, filtering, missing data treatment, and variable reduction. Additionally, I have learned several data visualization techniques.

For this project, I have chosen the healthcare measures data of adults and children for the United States of America in 2016. This data is issued by the health industry and covers almost every state of the USA. The dataset provides useful information on health measures such as the number of measures considered, the state rate on each measure, the location of the state, the number of state reports, and more. I have cleaned the data by removing unnecessary details and handling missing values using the Pandas library for data manipulation in Python. I have also used the Matplotlib library to create various graphs and visualizations depending on the type of data to gain different insights and identify patterns in the data, which can assist us in making predictions and decisions.

ANALYSIS OBJECTIVES

1. Find how many states included in health care measures?
2. Find the average state rate?
3. How many types of measures are taken into count?
4. How many states have more the 40 reporting in domain of Primary Care Access and Preventive Care?
5. Find how many reporting programs contain child core set?
6. Group the data by top and bottom quartile?
7. How many times Alabama State Occurs in data set?
8. Plot pie chart showing state rate for access to primary care practitioners for some state?
9. Plot bar chart showing state rate for access to primary care practitioners for some state?
10. Plot scatter plot showing state rate for access to primary care practitioners for some state?

DATA ACQUISITION AND CLEANING

Code to read the data from Excel / CSV / HTML.

```
healthData = pd.read_csv("adult_child_2016.csv")
```

Clean the unnecessary data, by removing, replace the missing data and renaming the columns.

```
#DATA CLEANING
#Missing value is filled by mean of its column
healthData['State Rate'].fillna(healthData['State Rate'].mean(),
inplace=True)

#print(healthData['State Rate'].head(5))
#rename FFY column as it is not abbreviated by a new person
healthData.rename(columns={'FFY': 'Year'},inplace=True)
print(healthData.columns)
```

```
#drop column that is irrelevant to our project
del healthData['Notes']
print(healthData.columns)
```

```
#drop column name measure Abbreviation
del healthData['Measure Abbreviation']
print(healthData.columns)
```

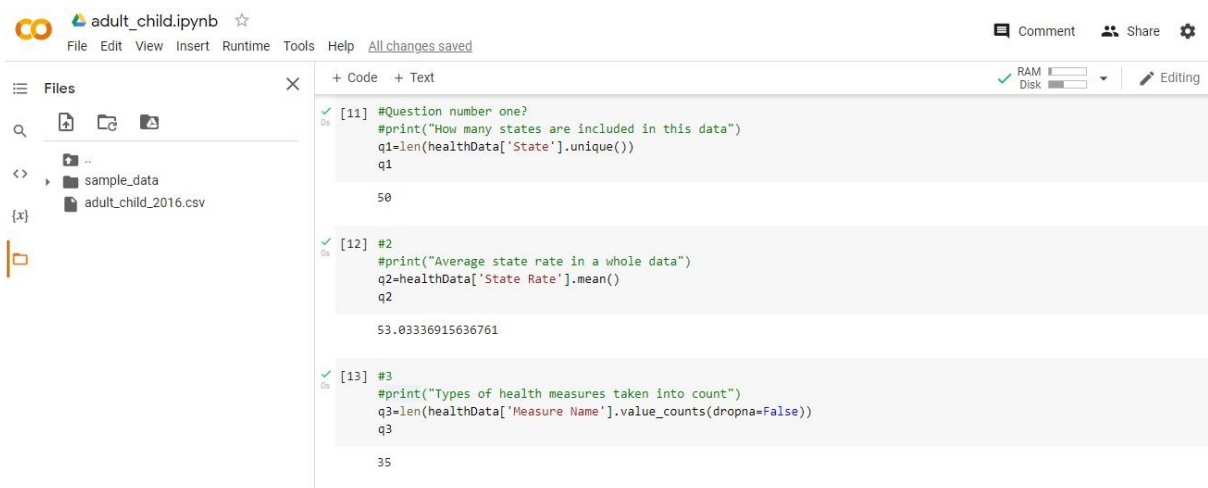
Explanation of why data clean needed (for your data).

- There are some values missing State Rate column which will affect our results so I fill those missing values with their mean value of State Rate column
- I also rename FFY column as year because FFY don't make any sense Year can easily be understand.
- There is column name Notes, the presence of which do not effect my data so I deleted that column.
- I also deleted a column name Measure Abbreviation because there is also a column named as measure Name which also give same information.

DATA AND EXPLORATORY ANALYSIS

Code :

Question 1 to 3 screenshots



The screenshot shows a Jupyter Notebook interface with the following content:

```
adult_child.ipynb ☆
File Edit View Insert Runtime Tools Help All changes saved
RAM 100% Disk 100% Editing

Files
sample_data
adult_child_2016.csv

+ Code + Text

[11] #Question number one?
#print("How many states are included in this data")
q1=len(healthData['State'].unique())
q1
50

[12] #2
#print("Average state rate in a whole data")
q2=healthData['State Rate'].mean()
q2
53.03336915636761

[13] #3
#print("Types of health measures taken into count")
q3=len(healthData['Measure Name'].value_counts(dropna=False))
q3
35
```

Question 4 to 5

adult_child.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

RAM Disk Editing

Files

- bin
- boot
- content
- datalab
- dev
- etc
- home
- lib
- lib32
- lib64
- media
- mnt

```
[14] #4
#print("how many Number of states have no of reporting in domain of Primary Care Access and Preventiv Care")
stateRate80plus=healthData[(healthData['Number of States Reporting']>40)&(healthData['Domain']=="Primary Care Access and Preventiv Care")]
q4=len(stateRate80plus['State'].unique())
q4
48
```

```
#5
#print("Number of child core set Reporting ")
childset=healthData[healthData['Reporting Program']=="Child Core Set"]
q5=len(childset)
q5
1399
```

Question 6 to 7

adult_child.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

RAM Disk Editing

Files

- bin
- boot
- content
- datalab
- dev
- etc
- home
- lib
- lib32
- lib64
- media
- mnt
- opt
- proc
- python-apt
- root
- run
- sbin
- srv

```
#6
#print("Group data based on top and bottom quartile ")
q6=healthData.groupby(['Bottom Quartile', 'Top Quartile']).mean()
q6.head(5)
```

	Year	State Rate	Number of States Reporting	Median
Bottom Quartile	Top Quartile			
4.8	1.7	2016	7.529271	32 3.0
6.9	17.9	2016	12.500000	26 13.7
10.0	8.0	2016	13.116132	26 8.9
15.7	50.5	2016	36.595701	26 36.0
16.2	23.6	2016	19.908000	41 20.8

```
#7
#print("How many times Alabama state appers in data for health measures")
groupData=healthData.groupby(['State'])
q7=len(groupData.get_group('Alabama'))
q7
69
```

Disk 66.03 GB available

Activate Windows
Go to Settings to activate Windows

Question 8

adult_child.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

RAM Disk Editing

```
#8
# #plots using matplotlib
ploting=healthData.sort_values(['Bottom Quartile','Median', 'Top Quartile'],ascending=[False,False,False])
pieplot=ploting.head(15)
pie=pieplot.set_index('State',inplace=True)
pieplot["State Rate"].plot.pie(title=" The state rate for access to primary care practitioners ",autopct="%%.2f")
mt.pie(pieplot["State Rate"],labels=pie)
mt.show()
```

The state rate for access to primary care practitioners

State	Rate
Texas	6.74
Colorado	6.43
Indiana	6.70
New Jersey	6.82
Mississippi	6.74
West Virginia	6.60
Massachusetts	6.73
Nebraska	6.69
Maryland	6.80
Arkansas	6.34
Mississippi	6.90
State of Georgia	6.55
District of Columbia	6.65
Minnesota	6.69
New York	6.62
Georgia	6.62

Activate Windows
Go to Settings to activate Windows

Question 9

adult_child.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

RAM Disk Editing

```
[9] #9
Graph=healthData.sort_values(['Bottom Quartile','Median', 'Top Quartile'],ascending=[True,True,True])
antipsychotics=Graph.head(15)
antipsychotics.set_index('State',inplace=True)
antipsychotics["State Rate"].plot.bar(title="The state rate ")
mt.show()
```

The state rate

State	Rate
California	5
Alabama	2
West Virginia	38
Vermont	5
Colorado	7
Texas	2
Wyoming	7
South Carolina	2
Iowa	4
Florida	2
New York	2
Nevada	2
Georgia	2
Tennessee	2
Arkansas	2

Activate Windows
Go to Settings to activate Windows

Question 10

adult_child.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

RAM Disk Editing

```
[10] #10
x = healthData['State'].head(10)
y = healthData['Number of States Reporting'].head(10)
mt.xlabel("States")
mt.ylabel("State Reportings")
mt.scatter(x, y)
mt.show()
```

State Reportings

State	Number of States Reporting
Mississippi	38
North Carolina	45
Alaska	25
West Virginia	50
Rhode Island	34
Idaho	45
New Jersey	41
Louisiana	46
States	33

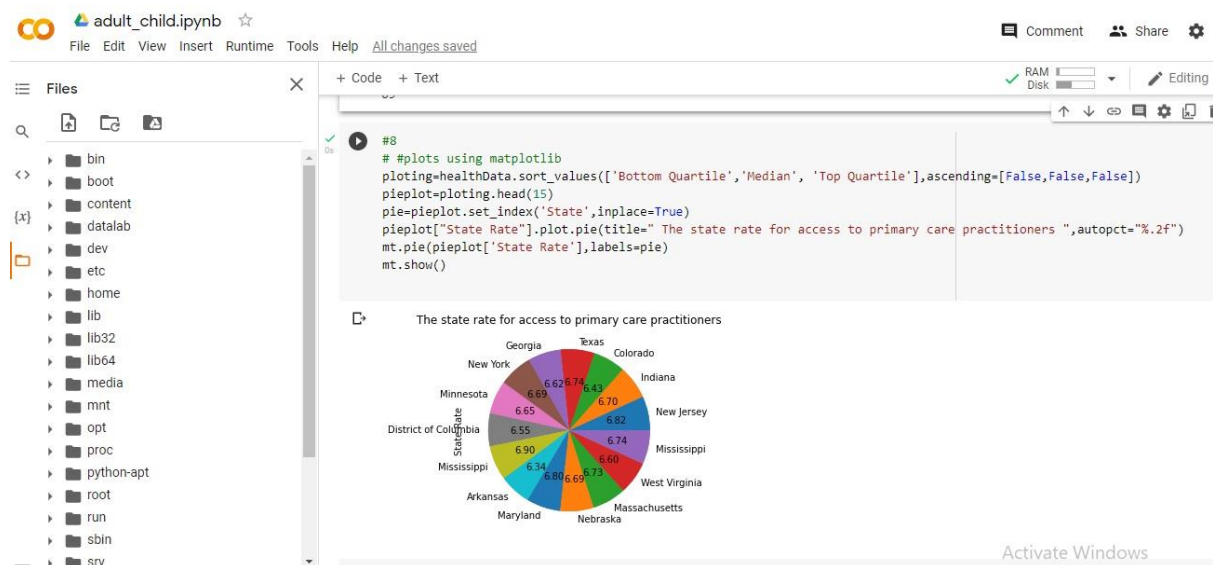
Activate Windows
Go to Settings to activate Windows

Explanation:

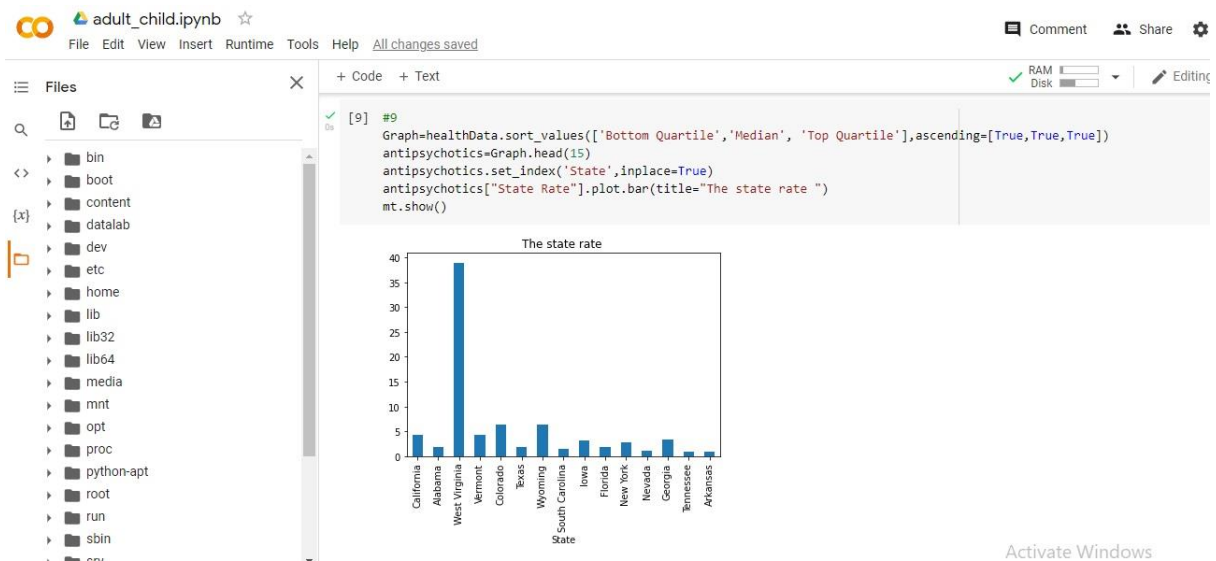
- In the above code, I have performed various operations on the dataset. Firstly, I used the Pandas .unique() method to find the number of states present in the dataset, and then applied the len() method to find the total number of states.
- Secondly, I calculated the average state rate by using the Pandas mean() method, which finds the average or mean of the whole column.
- Thirdly, I found the total number of health measures by taking some insights first, and then applied the Pandas value_counts() method, over which I applied the len() method to find the number of measures.
- In the fourth operation, I found out how many states have a number of reporting greater than 40.
- In the fifth operation, I found the number of child core set reportings in our dataset.
- In the sixth operation, I grouped the data based on bottom and top quartiles.
- In the seventh operation, I found out how many times Alabama State appears as a whole in our dataset.
- In the eighth operation, I plotted a pie chart showing 15 states from the first 15 columns with their state rate. Additionally, there are two other graphs showing bar and scatter charts.
- In the ninth operation, I plotted a bar plot showing state rate for the first 15 states from the dataset.
- In the tenth operation, I plotted a scatter plot showing 10 states and their number of reporting.

DATA ANALYSIS – VISUALIZATION

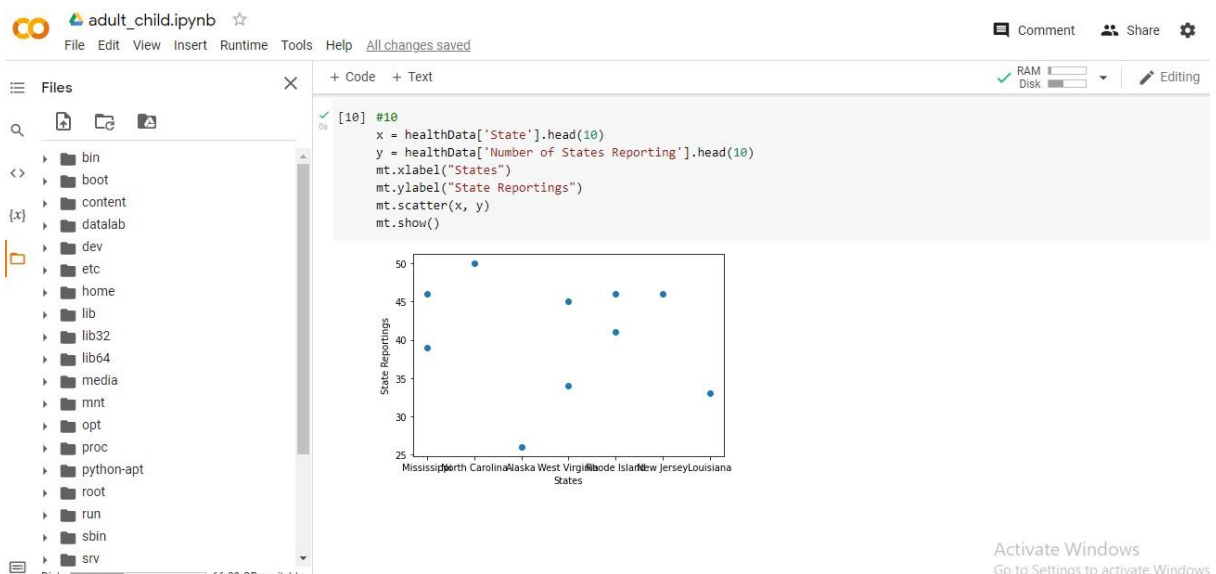
Pie chart:



Bar chart:



Scatter plot:



EXECUTIVE SUMMARY

In this project I have found different patterns in my selected data set, there are 1967 rows and 16 columns of my data set at first. Then I found that there are some columns which have some missing values, I fill those values with the mean of their column and then I found out that there are some useless columns like notes etc. I have deleted all those columns. So that my data gets cleaned. Then I also rename a column name FFY by Year as it will be easy to understand data.

Then I also remove another column name as measure abbreviation because there is another column named as measure name which also contains the same information. By doing all these steps I was clear and ready for insights or to draw different patterns for making different decisions. I have also drawn different plots like pie chart, bar chart, scatter plot, for getting different insights.

Recommendations:

1. I have found pandas and matplotlib as best tool for data cleaning and visualization respectively.
2. To avoid errors data organization and sorting is must.

For dealing in large datasets data visualization techniques are best e.g., Pie chart, Bar chart, scatter chart, Histogram etc.

REFERENCES

<https://www.oreilly.com/library/view/python-for-data/9781449323592/>