GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

PRESIDENCY UNIVERSITY

Private University Estd. in Karnataka State by Act No. 41 of 2013

KARANATAKA STATE BUDGET ANALYSIS

A Project Report Submitted By

(G2B1 batch-Group4)

B SAISUNIDHI (20181CSE0088)

B CHAITANYA KRISHNA (20181CSE0096)

B VEERANJANEYULU (20181CSE0099)

C H POOJASATHVIKA (20181CSE0121)

C MAMATHA (20181CSE0122)

as part of lab-based course Data Visualization, CSE 367

of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

under the supervision of

Mr. Deepak S Sakkari-Assistant Professor(CSE)
Ms. Shet Reshma-Assistant Professor (CSE)

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

PRESIDENCY UNIVERSITY

Private University Estd. in Karnataka State by Act No. 41 of 2013

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

BONAFIDE CERTIFICATE

Group-4 members :

B SAISUNIDHI (20181CSE0088)

B CHAITANYA KRISHNA (20181CSE0096)

B VEERANJANEYULU (20181CSE0099)

C H POOJASATHVIKA (20181CSE0121)

C MAMATHA (20181CSE0122)

ACKNOWLEDGEMENT

“It is not possible to prepare a project report without the assistance and encouragement of other people. This one is certainly no exception.”

On the very outset of this report, I would like to extend my sincere and heartfelt obligation towards all the personages who have helped me in this endeavor. Without their active guidance, help, cooperation and encouragement, I would not have made headway in the project.

I am ineffably indebted to my faculties Deepak S Sakkari - Asst . Prof-CSE and Shet Reshma -Asst . Prof-CSE for conscientious and valuable guidance, encouragement and support to accomplish this assignment and completion of this project in it presently.

I extend my gratitude to PRESIDENCY UNIVERSITY for giving me this opportunity.

I also acknowledge with a deep sense of reverence, my gratitude towards my parents and member of my family, who has always supported me morally as well as economically.

At last, but not least gratitude goes to all of my friends who directly or indirectly helped me to complete this project report.

Any omission in this brief acknowledgement does not mean lack of gratitude.

Thank you

CONTENTS

INTRODUCTION

The Global Burden of Diseases ,injuries and risk factors study(GBO) in 2019 found that police conflict and execution accounted for 293000 global deaths (95% uncertainty interval[UI]21500-34400)from 1980 to 2019.In recent years, particularly since the fatal shooting of Michael Brown in Ferguson ,Missouri in 2014 .In 2019 , the USA accounted for 13.2% (95% UI 11.6-15.1) of the global deaths (7710-9930)due to police conflict while only accounting for 4% of the global population; police conflict and executions was the estimated cause of death for 1150 deaths (998-1310) in the USA . The burden of police violence fatalities in the USA is known to fall disproportionately on Black, Indigenous and Hispanic population.

Apart from the obvious direct harm to victims, whose lives are cut short, there is the harm to families and communities that lose children, partners and parents. Public health has long declared violence a public health issue, but it has focused mainly on interpersonal violence.

Systemic and direct racism, manifested in laws and policies as well as personal implicit biasis, result in black, Indigenous and Hispanic Americans being the targets of police violence.

ABOUT DATASET:

- We used a dataset with a name shootings.csv from kaggle website.
- There are 4895 rows and 15 columns in this datset.
- Columns : Id , name , date , manner_of_death , armed ,age ,gender , race, city, state, signs_of_illness, threat_level , flee, body_camera , arms_category.
- ID : Represents the count of rows.
- Name : It indicates the person name who has dead or injured .
- Date : This column has not disclosed any of the information regarding the incident that has happened .Hence we remove this column during code execution.
- Manner_of_death : This column represents how the person is injured which caused the death .
  Example : shot or tasere etc..,
- Armed : This column gives the information about through which weapon the attacker attacked the opposite person or enemy .
  Example :Gun , toy weapon , knife etc..,
- Age : This column represents age of the person .
- Gender : Indicates the gender of person.
- Race : Represent the race of the person.
  Example: Asian, white, Hispanic, black etc..,
- City : It represent in which city this incident has occurred .
  Example: Shelton , Aloha ,Freeport etc.,,
- State : It represent in which state this incident has occurred.
  Example: CA, AZ etc..,
- Signs_of_illness : This column represents whether the person gone through any depression or any issues regarding health due to attack .
- Threat_level : It indicates in which the way used to harm the other person . Like whether through attack or some other .
- Flee : It Indicates during the attack wheather the person has escaped from danger or during attack .
- Body_camera : It represents whether the incident is captured or not . It shows whether the incident happened in public places in which the street cameras are present .
- Arms_category : It represent the which object are used during attack .
  Example : gun, sharp objects , blunt instruments etc..,

Python libaries used :

1. **Pandas :**

   It is a widely used data analysis and manipulation library in python

2. **Matplotlib :**

   Graphical plotting and for data visualization purpose.

3. **Seaborn :**

   It is a python based data visualization library based on matplotlib.it provides a high-level interface for drawing attractive and informative statistical information

Plots and graphs used in project:

1. **Heat map:**

   It shows the null values in data set

2. **Pie chart:**

   Pie chart is a circular graph that presents values as a proportionally slides, in our project, we used pie chart to display sectors with respect to attack and illness of people.

3. **Bar plot:**

   Bar chart displays categorical data with rectangular bars whose length or height corresponds to the value of each data point, in our project we use bar plot to display change with respect to sectors.

4. **Distplot:**

   Represents the univariate distribution of data i.e. data distribution of a variable against the density distribution.

5. **Boxen plot:**

   A box plot displays median, mean or maximum and minimum, outliers may be plotted as individuals

Code:

importing Libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set(rc ={‘figure.figsize’:( 11.7 , 8.27)})
import plotly.express as px
import plotly.graph_objects as go
import plotly.figure_factory as ff
from plotly.colors import n_colors
from plotly.subplots import make_subplots
```

1)To mount the google colab the email

```
#to mount
from google.colab import drive
drive.mount(‘/content/drive’)
```

2)To read the csv file

```
#import the csv file in dataframe format
data=pd.read_csv(‘/content/drive/Mydrive/Data/shootings.csv’)
```

3)to display the first 5 rows in the data set

```
data.head()
```

4)To display the last 5 rows in the data set

```
data.tail(5)
```

5)To print the descriptive statistics of the dataframe

```
data.describe()
```

6)To check with there is any duplicates values in data set

```
data.duplicated().any()
```

7)To find the rows and columns of the dataframe

```
data.shape
```

8)To find the information of the dataframe

```
Data.info()
```

- **Frequency of reason of death**

```
df=data[‘manner_of_death’].value_counts().reset_index().rename(columns={‘in
dex,:’manner_of_death’,’manner_of_death’:’count’})
fig = go.Figure(go.Bar(
      x=df[‘manner_of_death’],y=df[‘count’],marker_color=’purple’))
fig.update_layout(title_text=’frequency of reason of death’
,xaxis_title=”manner of death “,yaxis_title=”count”,height =500,width=500)
fig.show()
```

- **Weapons used most frequently**

```
Df=data[‘armed’].value_counts().reset_index().rename(columns={‘index’:’weapo
ns used’ ,’armed’:’count’})
fig = go.Figure(go.Bar(
      X=df[‘weapons used’],y=df[‘count’],maker_color=#7b6de3’))
fig.update_layout(title_text=’Frequency of different weapons used’
,xaxis_title=”weapons used” , yaxis_title=”count” ,height =700,width=1000)
fig.show()
```

- ### Gender Ratio

```
Df=data[‘gender’].value_counts().reset_index().rename(columns={‘index’:’gend
er’ ,’gender’:’count’})
fig = go.Figure(go.Bar(
      X=df[‘gender’],y=df[‘count’],maker_color=#148edb’))
fig.update_layout(title_text=’gender ratio’ ,xaxis_title=”gender”
,yaxis_title=”count” ,height =700,width=500)
fig.show()
```

- ### Frequency of different Race

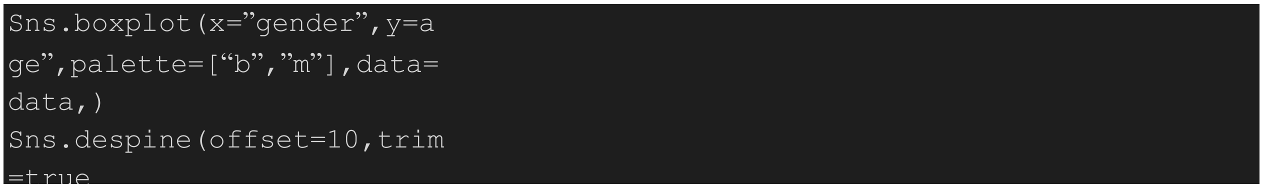
```
f=data[‘race’].value_counts().reset_index().rename(columns={‘index,:’race’,’r
ace’:’count’})
fig = go.Figure(go.Bar(
      x=df[‘race’],y=df[‘count’],
      marker={‘color’:df[‘count’],
      ‘colorscale’:’viridis’ },))
fig.update_layout(title_text=’frequency of reason of race’
,xaxis_title=”race “,yaxis_title=”count”,height =500,width=500)
fig.show()
```

- ### Age Distribution

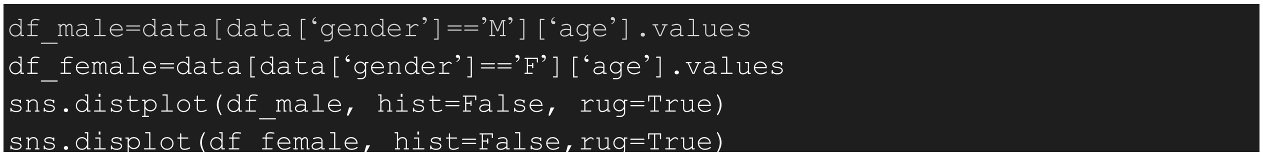
```
fig = go.Figure(go.Box(y=data[‘age’],name=”Age”))
fig.update_layout(title=”Distribution of Age”)
fig.show()
```

```
# DISTPLOT
Sns.distplot(data[‘age’],kde=True,color=’r’)
```

- **Age Distribution Male vs Female**

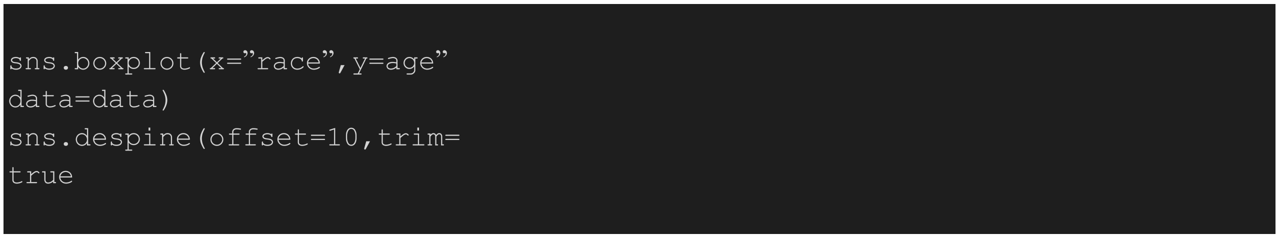


```
Sns.boxplot(x=”gender”,y=a
ge”,palette=[“b”,”m”],data=
data,)
Sns.despine(offset=10,trim
=true
```



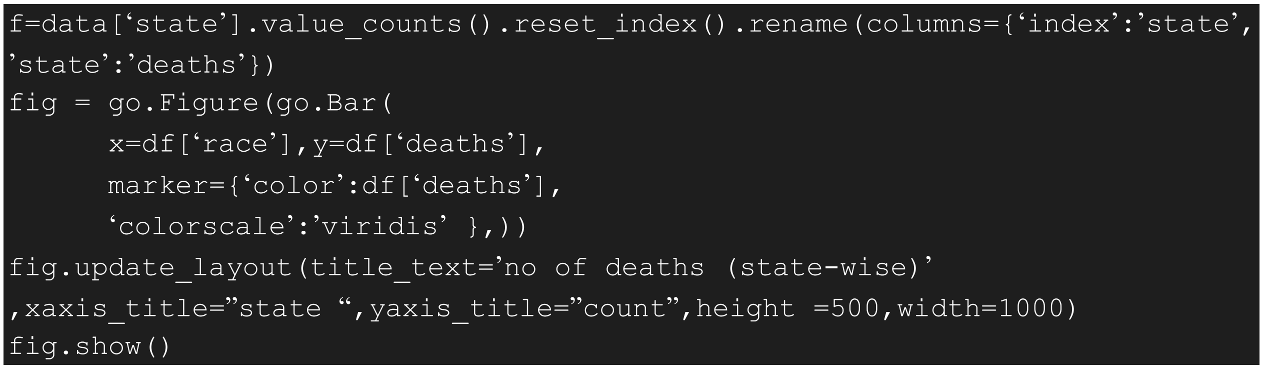
```
df_male=data[data[‘gender’]==’M’][‘age’].values
df_female=data[data[‘gender’]==’F’][‘age’].values
sns.distplot(df_male, hist=False, rug=True)
sns.displot(df_female, hist=False,rug=True)
```

- **Age Distribution of Different Race**



```
sns.boxplot(x=”race”,y=age”
data=data)
sns.despine(offset=10,trim=
true
```

- **Number of Deaths(State-Wise)**



```
f=data[‘state’].value_counts().reset_index().rename(columns={‘index’:’state’,
’state’:’deaths’})
fig = go.Figure(go.Bar(
      x=df[‘race’],y=df[‘deaths’],
      marker={‘color’:df[‘deaths’],
      ‘colorscale’:’viridis’ },))
fig.update_layout(title_text=’no of deaths (state-wise)’
,xaxis_title=”state “,yaxis_title=”count”,height =500,width=1000)
fig.show()
```

- **Signs of mental illness**

```
f=data[‘signs_of_mental_illnes’].value_counts().reset_index().rename(columns
={‘index’:’signs_of_mental_illness’,’signs_of_mental_illness’:’count’})
fig =
go.Figure(go.pie(lables=df[‘signs_of_mental_illness’],values=df[‘count’])])
fig.update_layout(title=”signs_of_mental_illness”,title_x=0.5)
fig.show()
```

```
df=data.groupby([‘data’,’gender’,’race’])[‘manner_of_death’].count().reset_ind
ex()
df[‘date’]=pd.to_datetime(df[‘date’])
df[‘year_month’]=df[‘date’].apply[lambda x: str(x.year))
df_ym=df.groupby([‘year_month’,’gender’,’race’])[[‘manner_of_death’]].sum().re
set_index()
df_ym[‘year_month’]=pd.to_datetime(df_ym[‘year_month’])
df_ym=df_ym.sort_values(‘year_month’)
df_ym[‘year_month’]=df_ym[‘year_month’].astype(‘str’).apply(lambda x:
x.split(‘-‘)[0])
```

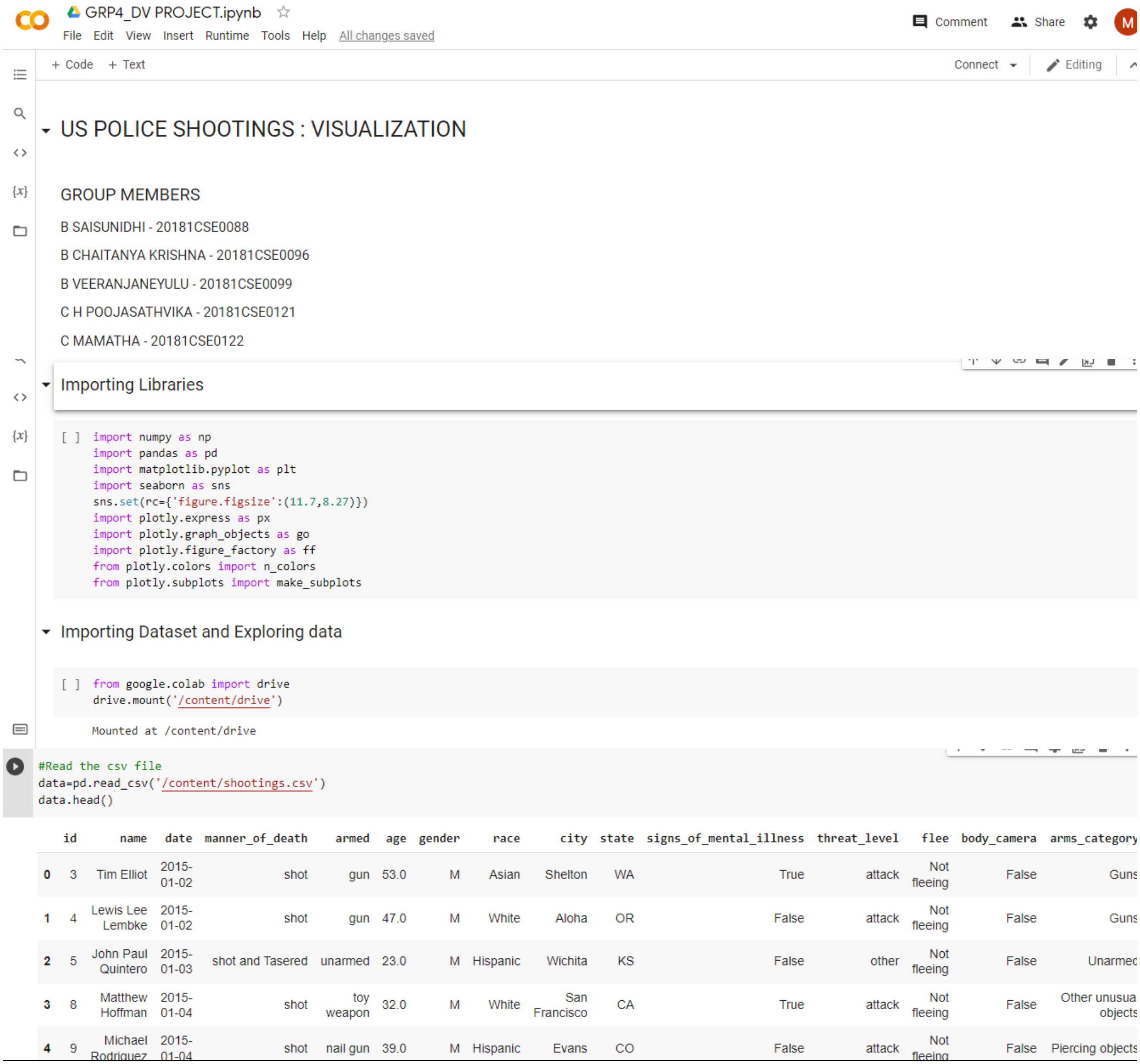
- **Threat level**

```
f=data[‘threat_value’].value_counts().reset_index().rename(columns={‘index’:’
threat_level’,’threat_level’:’count’})
fig = go.Figure(go.pie(lables=df[‘threat_level’],values=df[‘count’])])
fig.update_layout(title=”threat_level”,title_x=0.5)
fig.show()
```

- **Top 20 cities where most shooting occurred**

```
f=data[‘city’].value_counts().reset_index().rename(columns={‘index’:’city’,’ci
ty’:’deaths’})
sns.barplot(y=”city”,x=”deaths”,data=df,label=”death
s”)
```

OUTPUT SCREEN SHOTS:



US POLICE SHOOTINGS : VISUALIZATION

GROUP MEMBERS

B SAISUNIDHI - 20181CSE0088

B CHAITANYA KRISHNA - 20181CSE0096

B VEERANJANEYULU - 20181CSE0099

C H POOJASATHVIKA - 20181CSE0121

C MAMATHA - 20181CSE0122

Importing Libraries

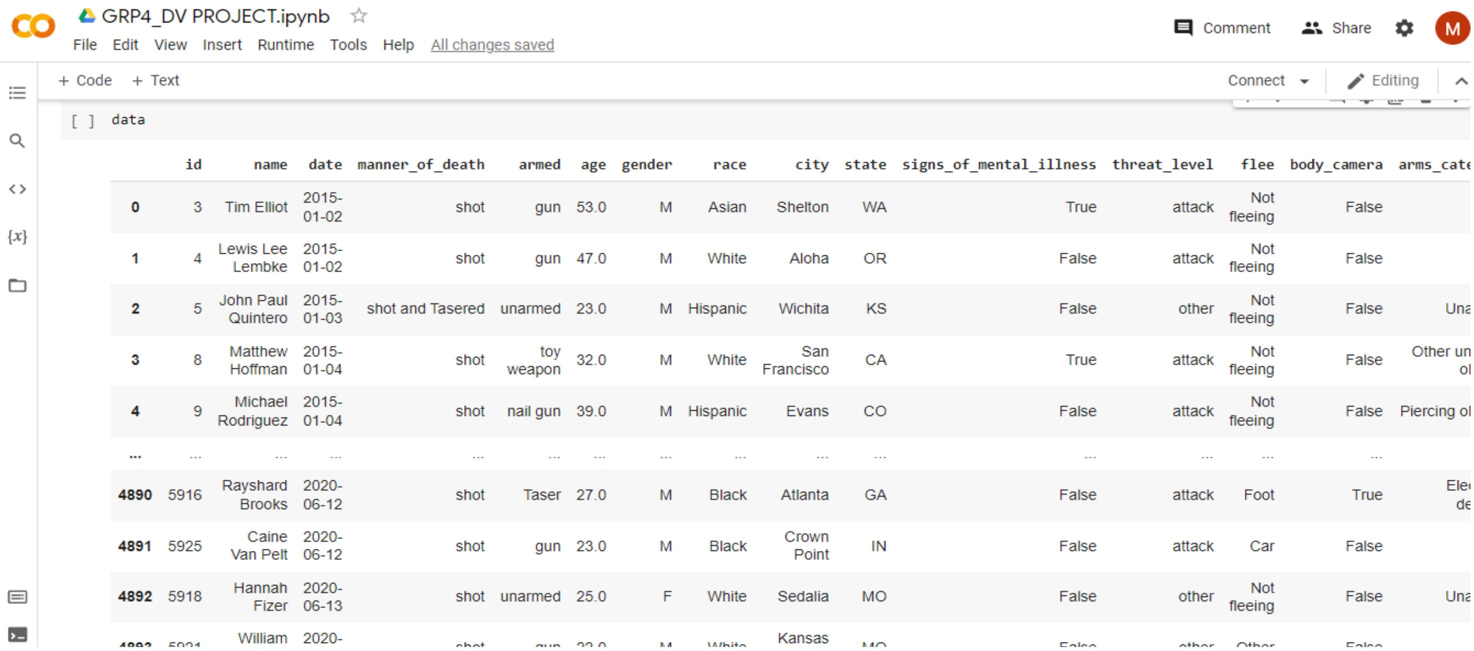
```
[ ] import numpy as np
    import pandas as pd
    import matplotlib.pyplot as plt
    import seaborn as sns
    sns.set(rc={'figure.figsize':(11.7,8.27)})
    import plotly.express as px
    import plotly.graph_objects as go
    import plotly.figure_factory as ff
    from plotly.colors import n_colors
    from plotly.subplots import make_subplots
```

Importing Dataset and Exploring data

```
[ ] from google.colab import drive
    drive.mount('/content/drive')

    Mounted at /content/drive
```

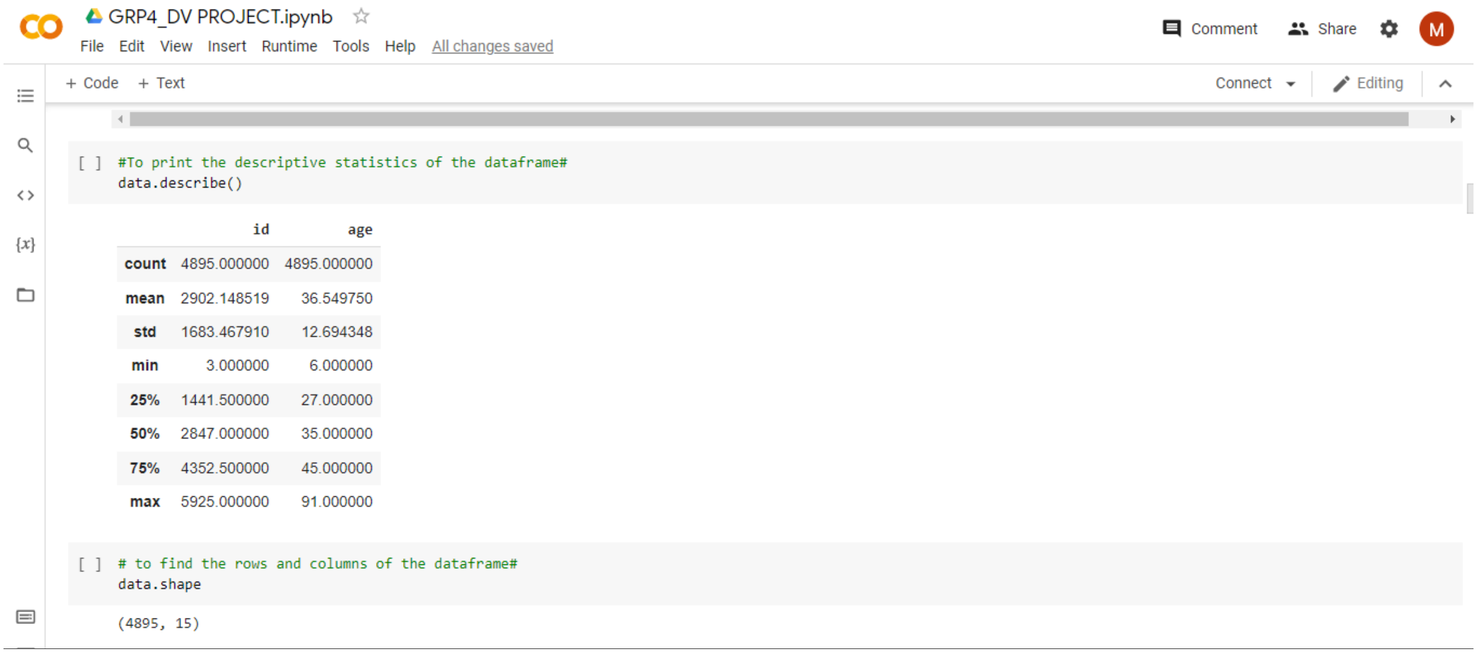
```
#Read the csv file
data=pd.read_csv('/content/shootings.csv')
data.head()
```

	id	name	date	manner_of_death	armed	age	gender	race	city	state	signs_of_mental_illness	threat_level	flee	body_camera	arms_category
0	3	Tim Elliot	2015-01-02	shot	gun	53.0	M	Asian	Shelton	WA	True	attack	Not fleeing	False	Guns
1	4	Lewis Lee Lembke	2015-01-02	shot	gun	47.0	M	White	Aloha	OR	False	attack	Not fleeing	False	Guns
2	5	John Paul Quintero	2015-01-03	shot and Tasered	unarmed	23.0	M	Hispanic	Wichita	KS	False	other	Not fleeing	False	Unarmec
3	8	Matthew Hoffman	2015-01-04	shot	toy weapon	32.0	M	White	San Francisco	CA	True	attack	Not fleeing	False	Other unusua objects
4	9	Michael Rodriguez	2015-01-04	shot	nail gun	39.0	M	Hispanic	Evans	CO	False	attack	Not fleeing	False	Piercing objects



```
[ ] data
```

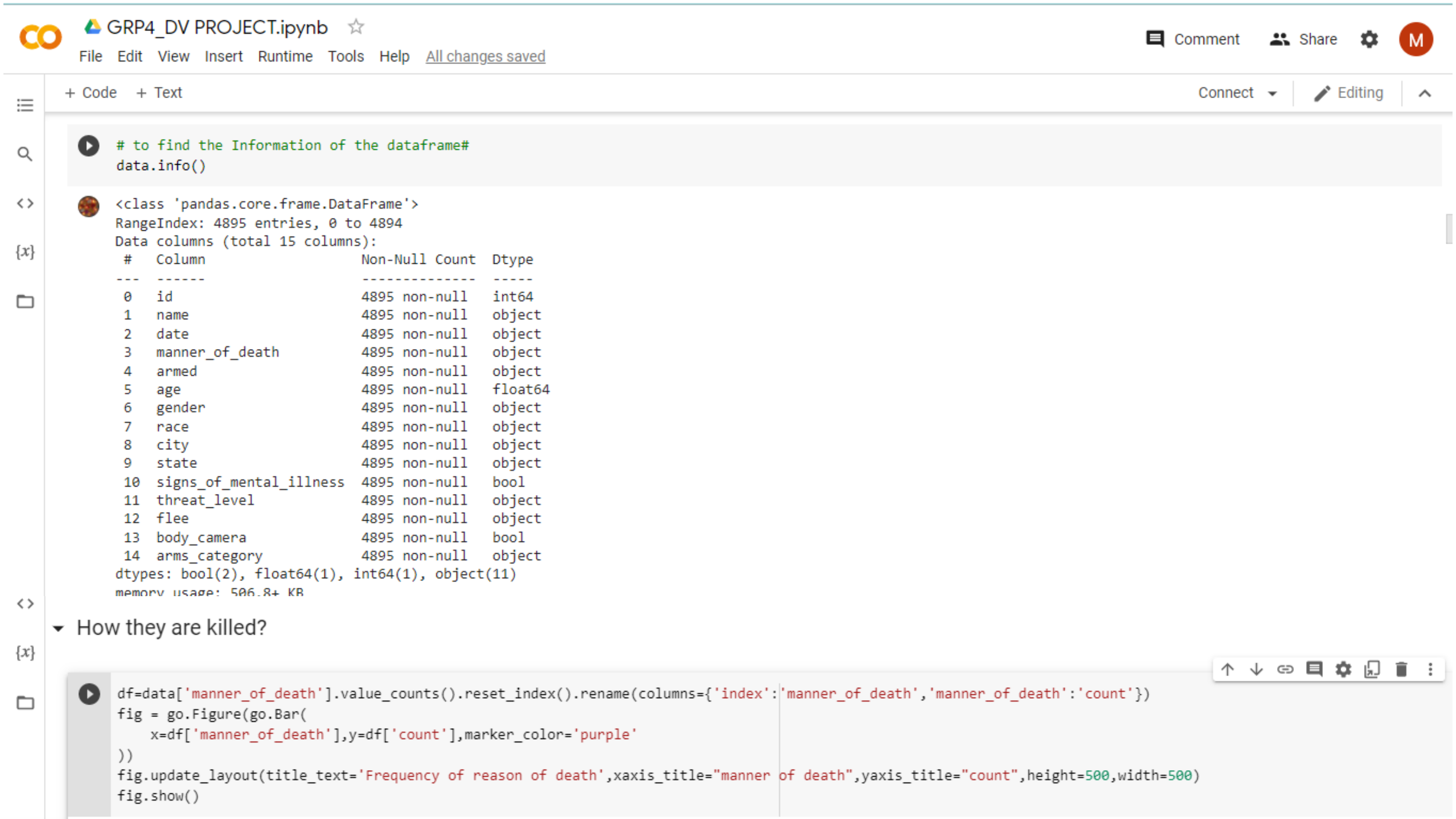
	id	name	date	manner_of_death	armed	age	gender	race	city	state	signs_of_mental_illness	threat_level	flee	body_camera	arms_cate
0	3	Tim Elliot	2015-01-02	shot	gun	53.0	M	Asian	Shelton	WA	True	attack	Not fleeing	False	
1	4	Lewis Lee Lembke	2015-01-02	shot	gun	47.0	M	White	Aloha	OR	False	attack	Not fleeing	False	
2	5	John Paul Quintero	2015-01-03	shot and Tasered	unarmed	23.0	M	Hispanic	Wichita	KS	False	other	Not fleeing	False	Una
3	8	Matthew Hoffman	2015-01-04	shot	toy weapon	32.0	M	White	San Francisco	CA	True	attack	Not fleeing	False	Other un ol
4	9	Michael Rodriguez	2015-01-04	shot	nail gun	39.0	M	Hispanic	Evans	CO	False	attack	Not fleeing	False	Piercing ol
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
4890	5916	Rayshard Brooks	2020-06-12	shot	Taser	27.0	M	Black	Atlanta	GA	False	attack	Foot	True	Ele de
4891	5925	Caine Van Pelt	2020-06-12	shot	gun	23.0	M	Black	Crown Point	IN	False	attack	Car	False	
4892	5918	Hannah Fizer	2020-06-13	shot	unarmed	25.0	F	White	Sedalia	MO	False	other	Not fleeing	False	Una
4893	5921	William	2020-	shot	gun	22.0	M	White	Kansas	MO	False	other	Other	False	



```
[ ] #To print the descriptive statistics of the dataframe#
    data.describe()
```

	id	age
count	4895.000000	4895.000000
mean	2902.148519	36.549750
std	1683.467910	12.694348
min	3.000000	6.000000
25%	1441.500000	27.000000
50%	2847.000000	35.000000
75%	4352.500000	45.000000
max	5925.000000	91.000000

```
[ ] # to find the rows and columns of the dataframe#
    data.shape

    (4895, 15)
```

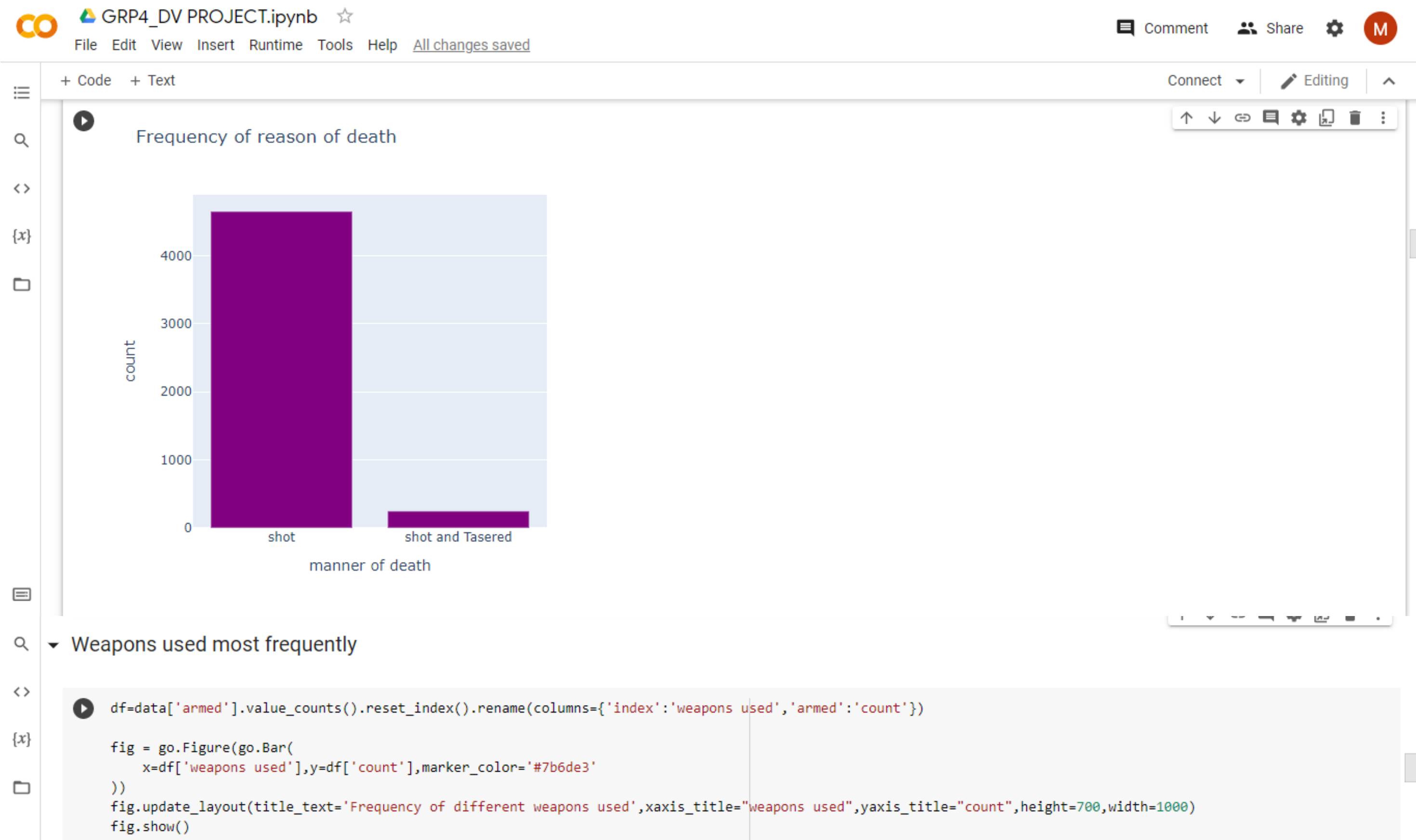


```
# to find the Information of the dataframe#
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4895 entries, 0 to 4894
Data columns (total 15 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   id                       4895 non-null   int64  
 1   name                     4895 non-null   object 
 2   date                     4895 non-null   object 
 3   manner_of_death          4895 non-null   object 
 4   armed                    4895 non-null   object 
 5   age                      4895 non-null   float64
 6   gender                   4895 non-null   object 
 7   race                     4895 non-null   object 
 8   city                     4895 non-null   object 
 9   state                    4895 non-null   object 
 10  signs_of_mental_illness  4895 non-null   bool   
 11  threat_level             4895 non-null   object 
 12  flee                     4895 non-null   object 
 13  body_camera              4895 non-null   bool   
 14  arms_category            4895 non-null   object 
dtypes: bool(2), float64(1), int64(1), object(11)
memory usage: 506.8+ KB
```

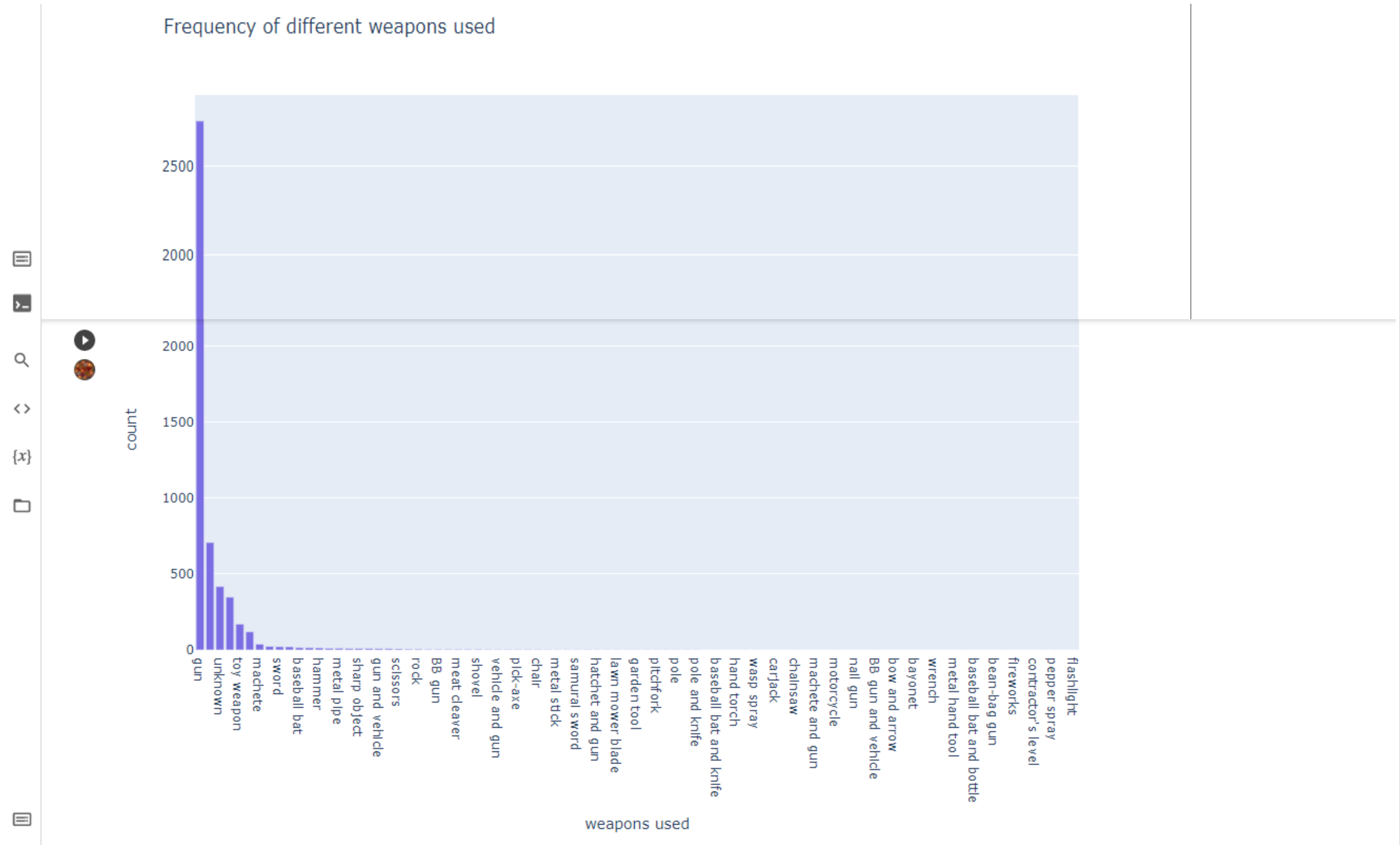
How they are killed?

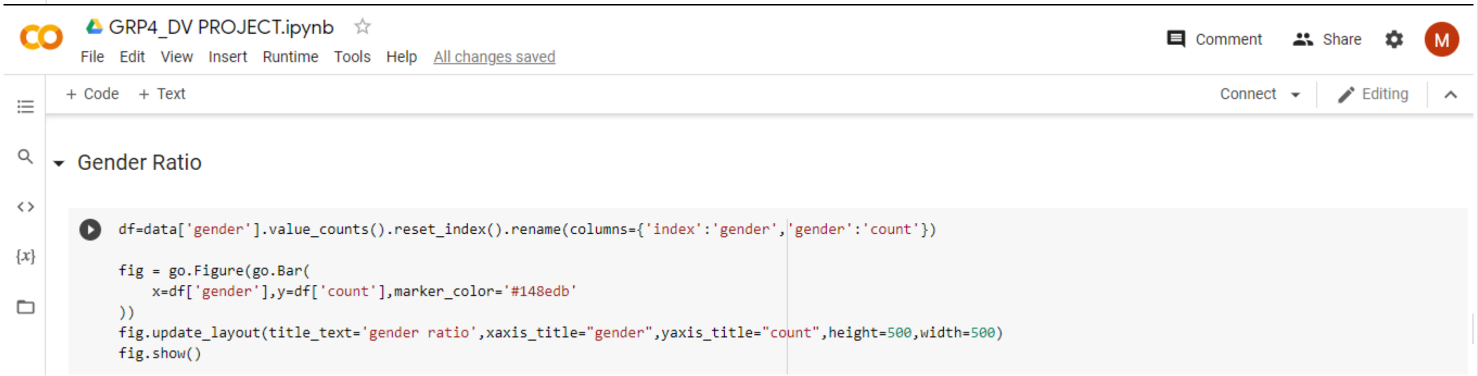
```
df=data['manner_of_death'].value_counts().reset_index().rename(columns={'index':'manner_of_death','manner_of_death':'count'})
fig = go.Figure(go.Bar(
    x=df['manner_of_death'],y=df['count'],marker_color='purple'
))
fig.update_layout(title_text='Frequency of reason of death',xaxis_title="manner of death",yaxis_title="count",height=500,width=500)
fig.show()
```



Weapons used most frequently

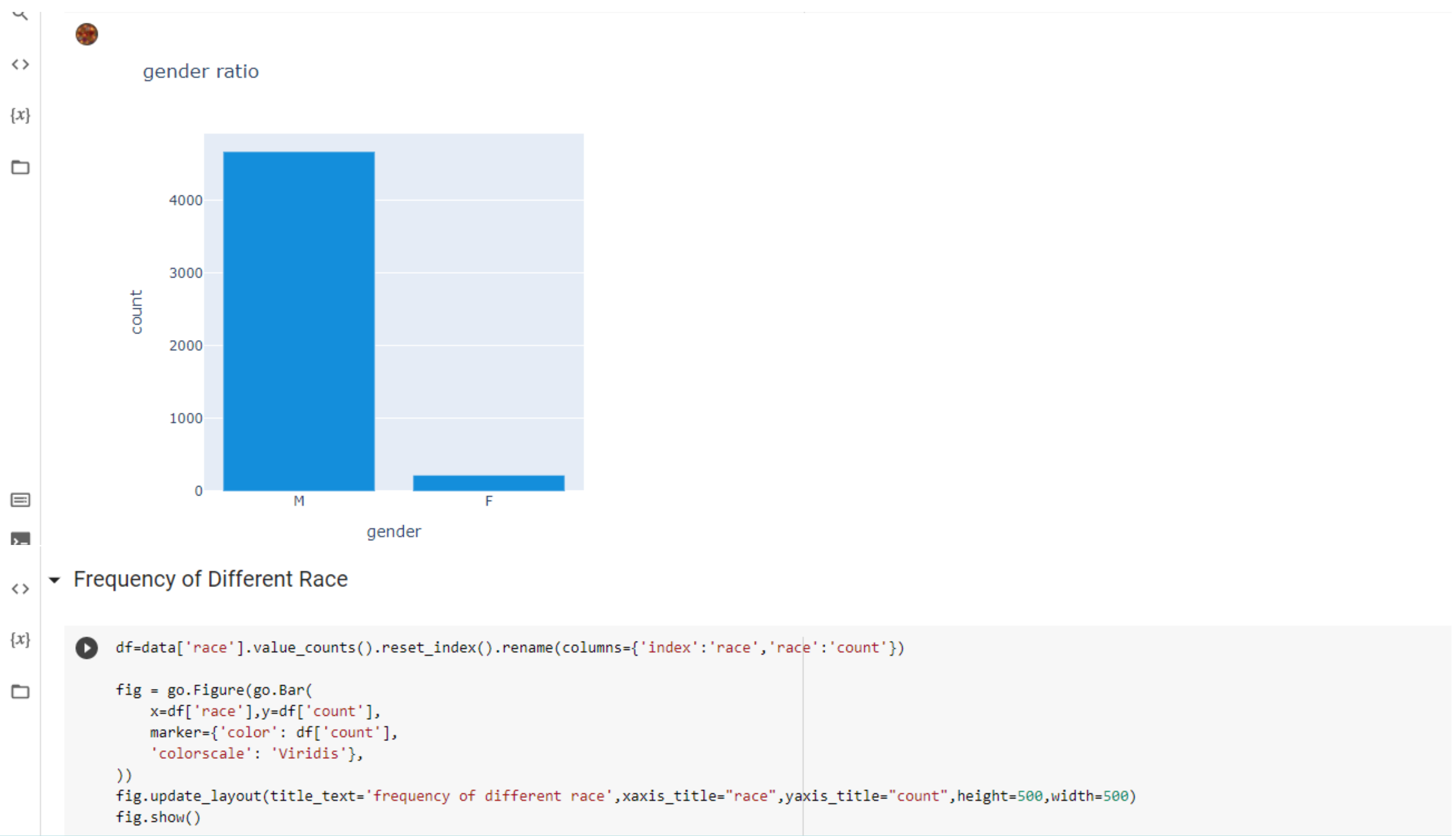
```
df=data['armed'].value_counts().reset_index().rename(columns={'index':'weapons used','armed':'count'})

fig = go.Figure(go.Bar(
    x=df['weapons used'],y=df['count'],marker_color='#7b6de3'
))
fig.update_layout(title_text='Frequency of different weapons used',xaxis_title="weapons used",yaxis_title="count",height=700,width=1000)
fig.show()
```





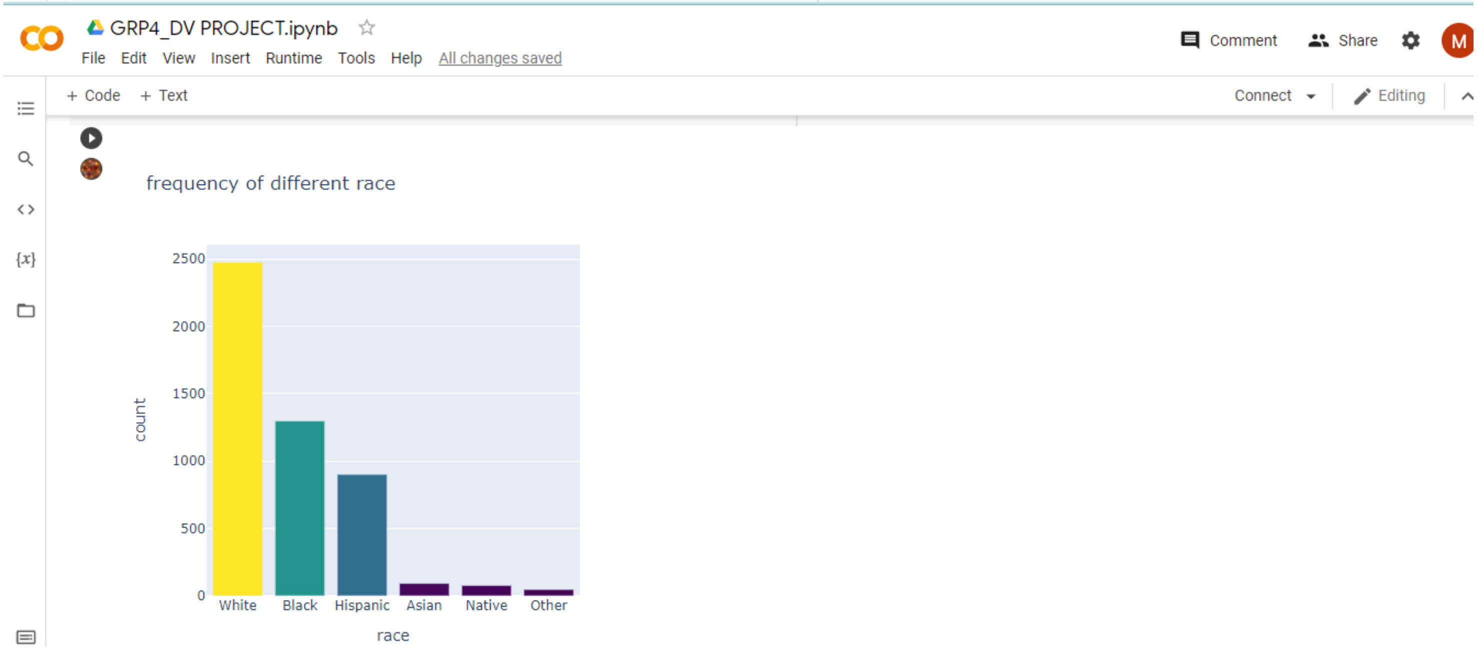
Gender Ratio

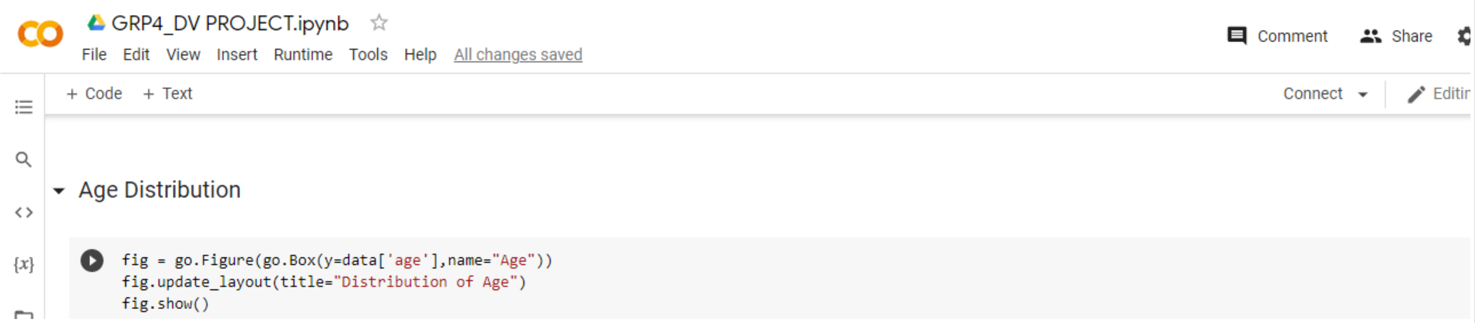
```
df=data['gender'].value_counts().reset_index().rename(columns={'index':'gender','gender':'count'})

fig = go.Figure(go.Bar(
    x=df['gender'],y=df['count'],marker_color='#148edb'
))
fig.update_layout(title_text='gender ratio',xaxis_title="gender",yaxis_title="count",height=500,width=500)
fig.show()
```



Frequency of Different Race

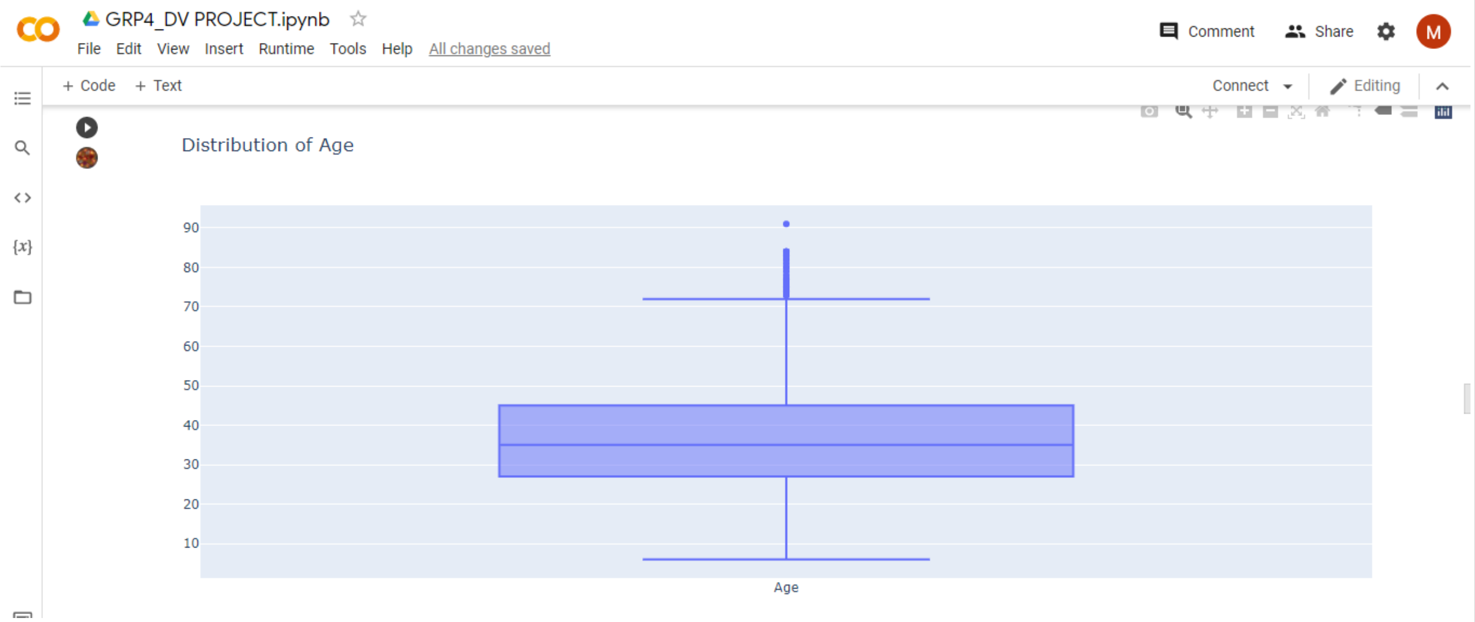
```
df=data['race'].value_counts().reset_index().rename(columns={'index':'race','race':'count'})

fig = go.Figure(go.Bar(
    x=df['race'],y=df['count'],
    marker={'color': df['count'],
    'colorscale': 'Viridis'},
))
fig.update_layout(title_text='frequency of different race',xaxis_title="race",yaxis_title="count",height=500,width=500)
fig.show()
```

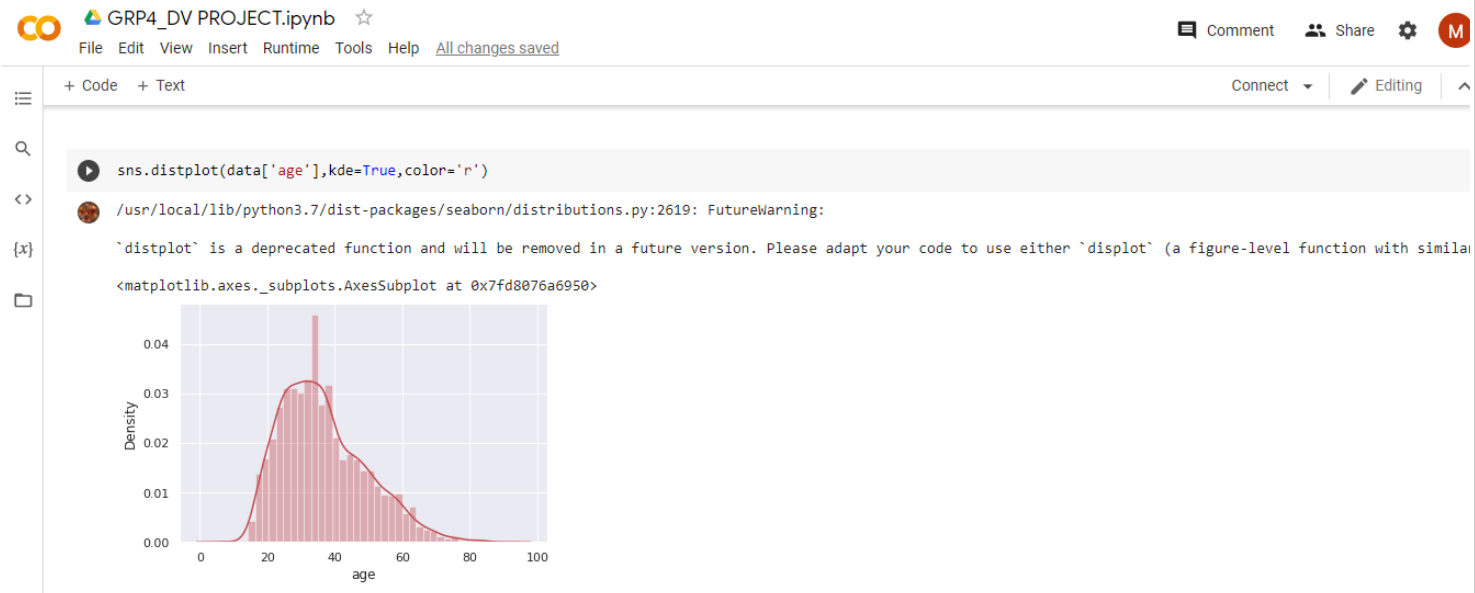




Age Distribution

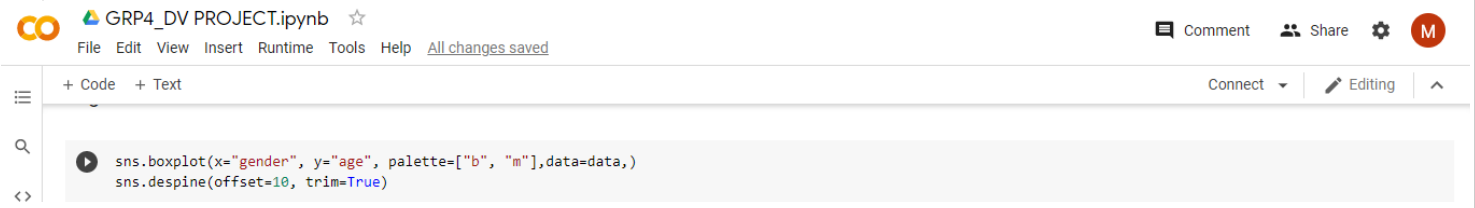
```
fig = go.Figure(go.Box(y=data['age'],name="Age"))
fig.update_layout(title="Distribution of Age")
fig.show()
```



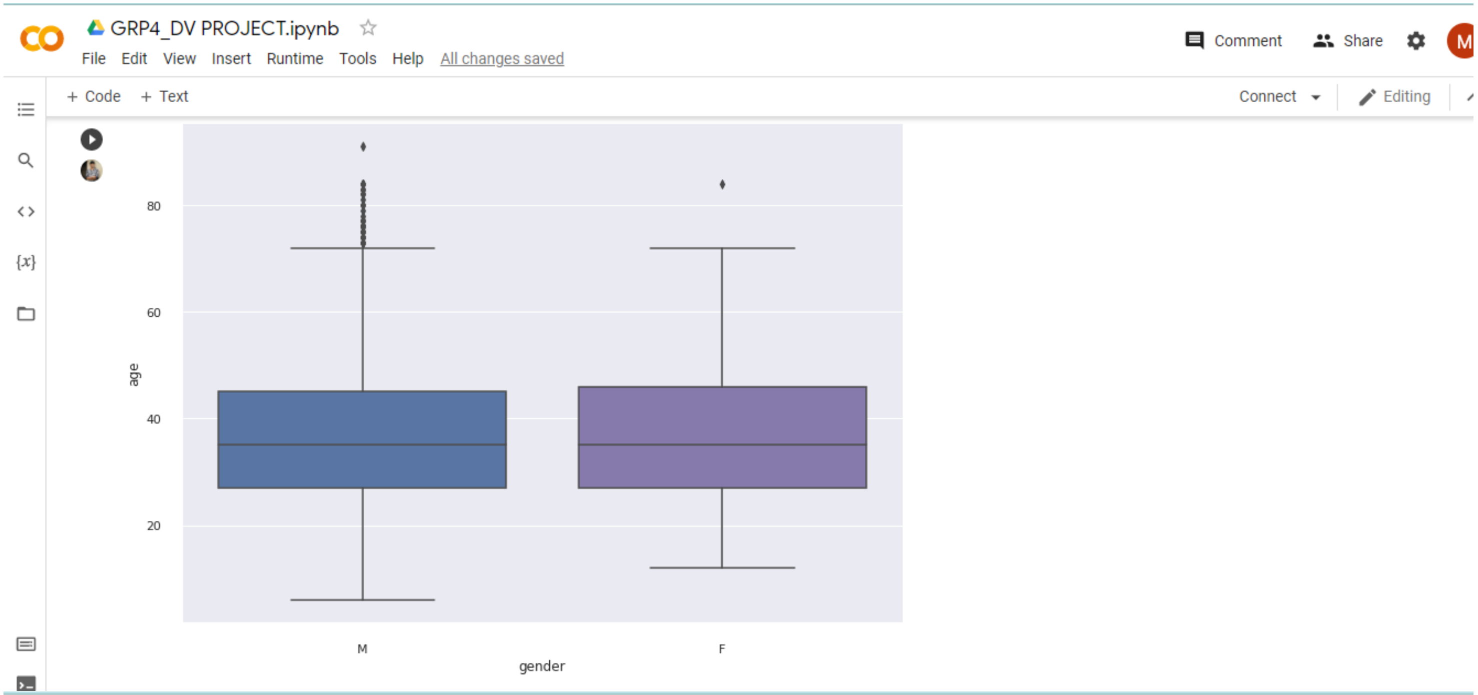


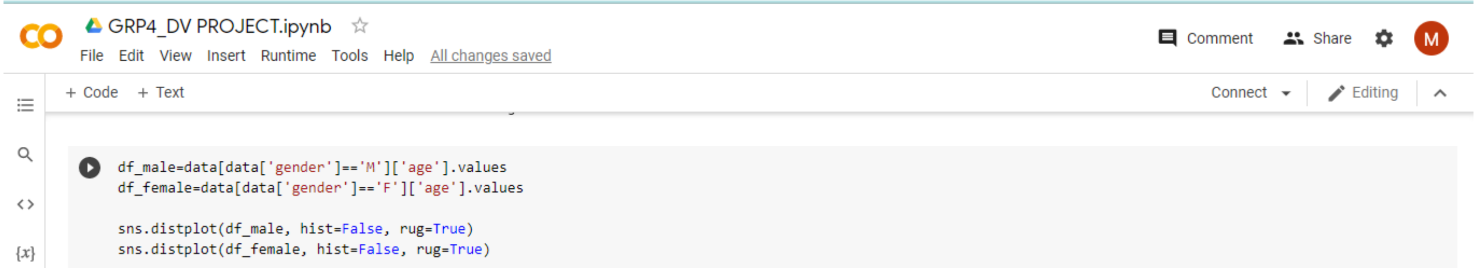
```
sns.distplot(data['age'],kde=True,color='r')
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning:
`distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with simila
<matplotlib.axes._subplots.AxesSubplot at 0x7fd8076a6950>
```

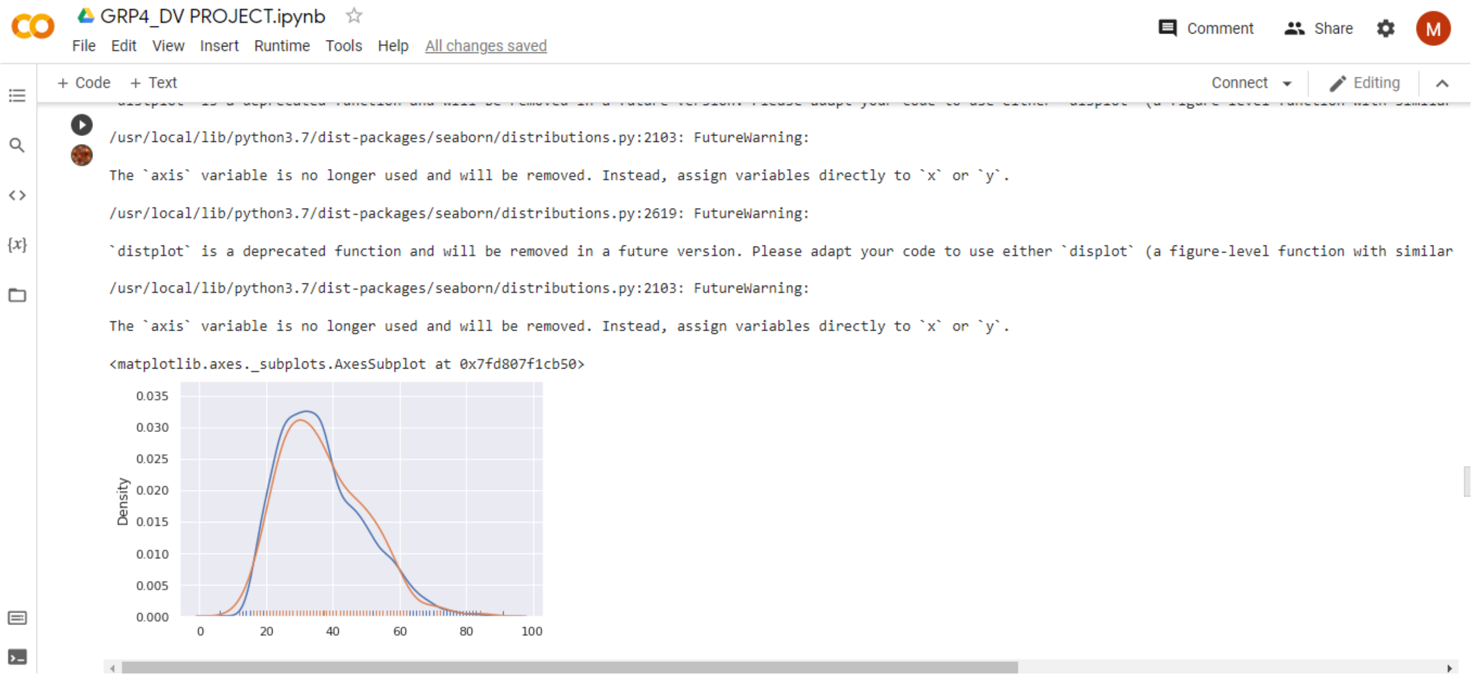


```
sns.boxplot(x="gender", y="age", palette=["b", "m"],data=data,)
sns.despine(offset=10, trim=True)
```

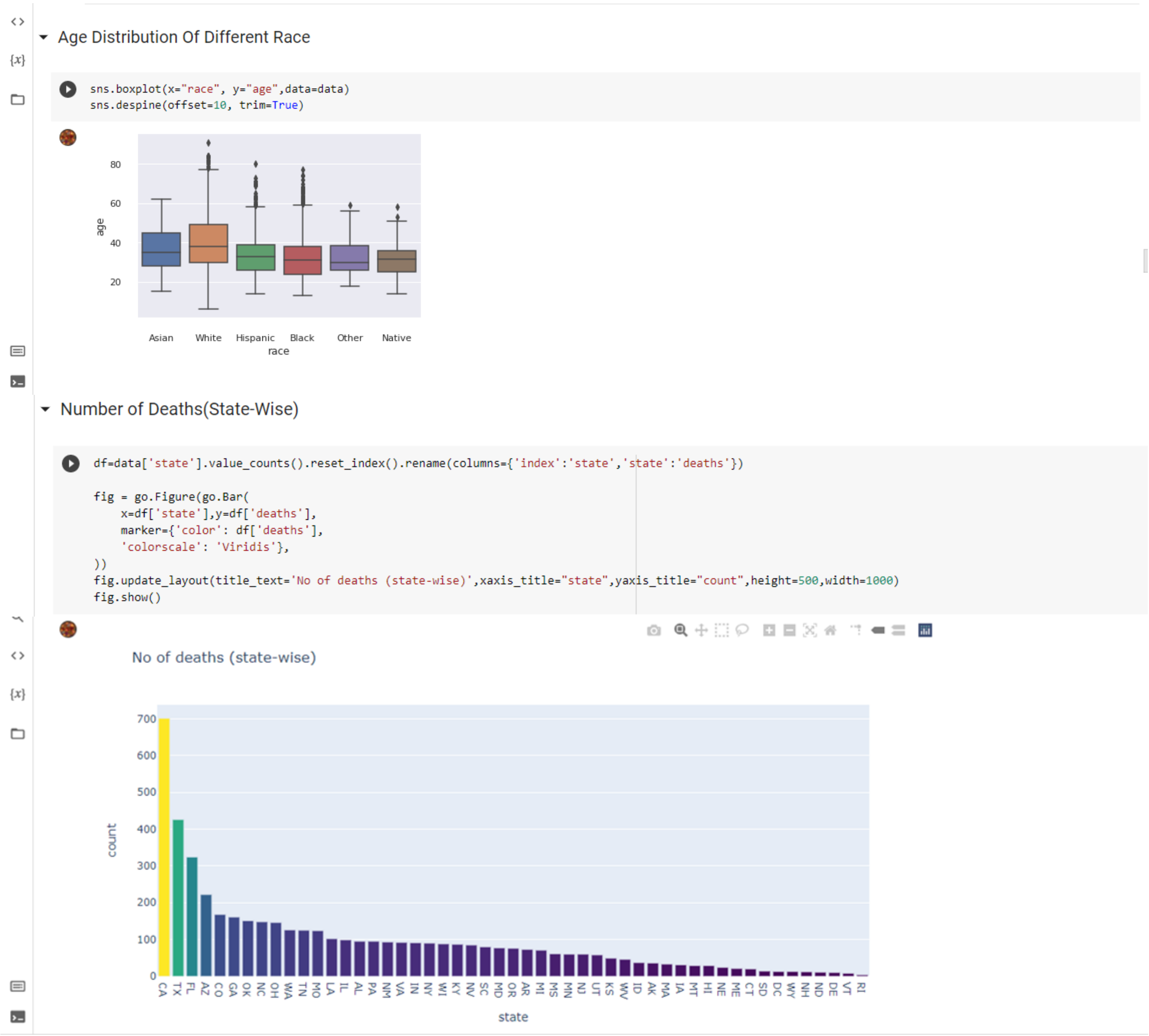




```
df_male=data[data['gender']=='M']['age'].values
df_female=data[data['gender']=='F']['age'].values

sns.distplot(df_male, hist=False, rug=True)
sns.distplot(df_female, hist=False, rug=True)
```



```
`distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2103: FutureWarning:

The `axis` variable is no longer used and will be removed. Instead, assign variables directly to `x` or `y`.

/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning:

`distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar

/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2103: FutureWarning:

The `axis` variable is no longer used and will be removed. Instead, assign variables directly to `x` or `y`.

<matplotlib.axes._subplots.AxesSubplot at 0x7fd807f1cb50>
```



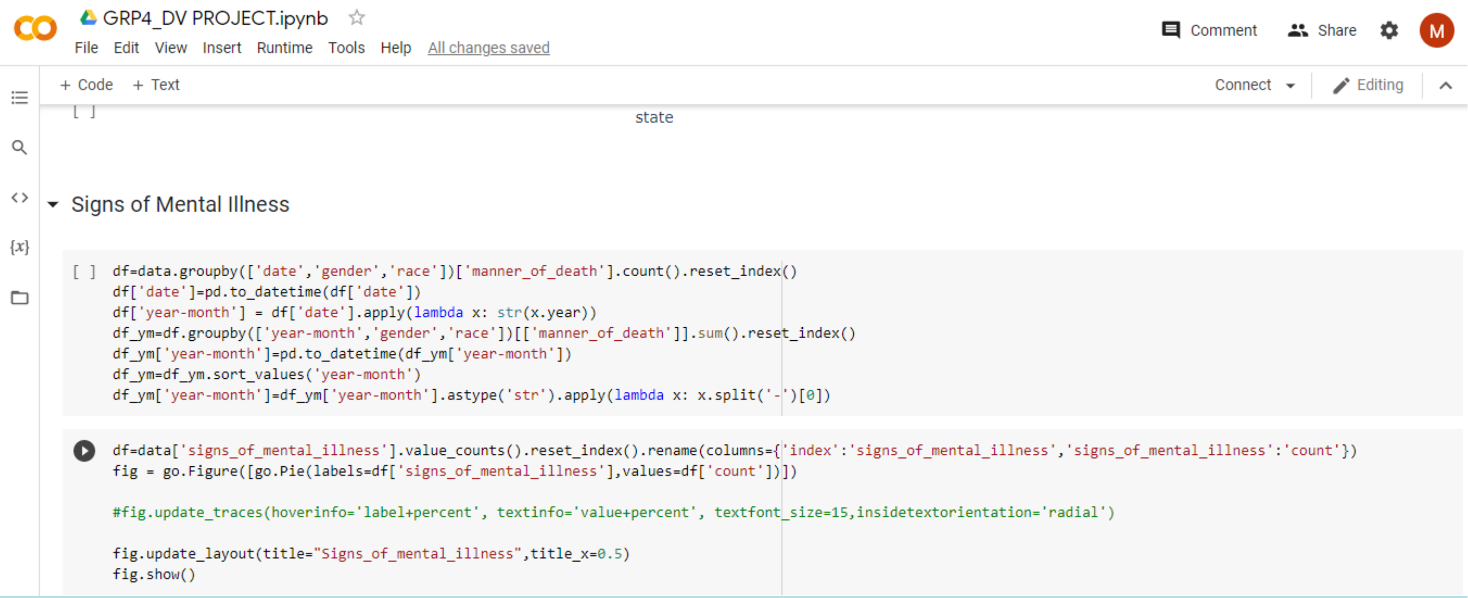
Age Distribution Of Different Race

```
sns.boxplot(x="race", y="age",data=data)
sns.despine(offset=10, trim=True)
```

Number of Deaths(State-Wise)

```
df=data['state'].value_counts().reset_index().rename(columns={'index':'state','state':'deaths'})

fig = go.Figure(go.Bar(
    x=df['state'],y=df['deaths'],
    marker={'color': df['deaths'],
    'colorscale': 'Viridis'},
))
fig.update_layout(title_text='No of deaths (state-wise)',xaxis_title="state",yaxis_title="count",height=500,width=1000)
fig.show()
```

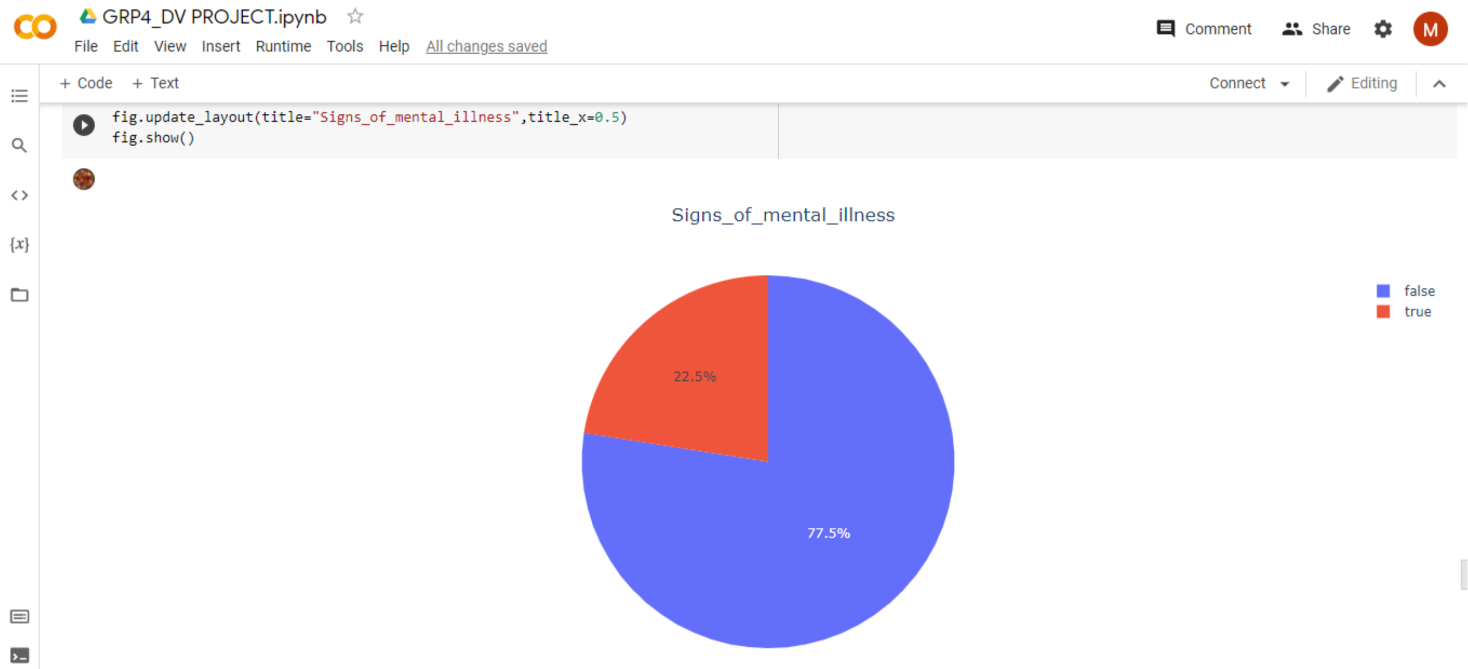
No of deaths (state-wise)
700
600
500
400
count
300
200
100
0
CA TX FL AZ CO GA OK NC OH WA TN MO LA IL AL PA NM VA IN NY WI KY NV SC MD OR AR MI MS MN NJ UT KS WV ID AK MA IA MT HI NE ME CT SD DC WY NH ND DE VT RI
state



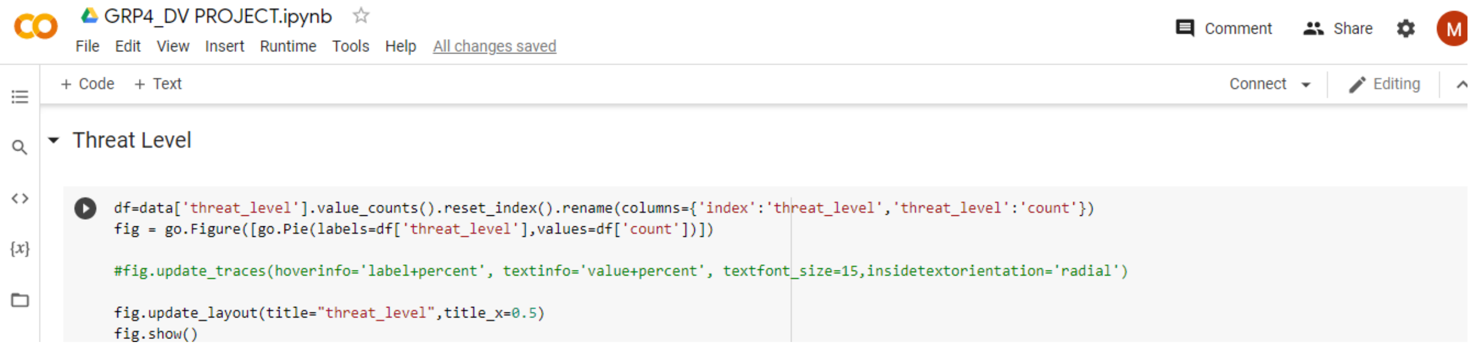
Signs of Mental Illness

```
[ ] df=data.groupby(['date','gender','race'])['manner_of_death'].count().reset_index()
    df['date']=pd.to_datetime(df['date'])
    df['year-month'] = df['date'].apply(lambda x: str(x.year))
    df_ym=df.groupby(['year-month','gender','race'])[['manner_of_death']].sum().reset_index()
    df_ym['year-month']=pd.to_datetime(df_ym['year-month'])
    df_ym=df_ym.sort_values('year-month')
    df_ym['year-month']=df_ym['year-month'].astype('str').apply(lambda x: x.split('-')[0])
```

```
df=data['signs_of_mental_illness'].value_counts().reset_index().rename(columns={'index':'signs_of_mental_illness','signs_of_mental_illness':'count'})
fig = go.Figure([go.Pie(labels=df['signs_of_mental_illness'],values=df['count'])])

#fig.update_traces(hoverinfo='label+percent', textinfo='value+percent', textfont_size=15,insidetextorientation='radial')

fig.update_layout(title="Signs_of_mental_illness",title_x=0.5)
fig.show()
```

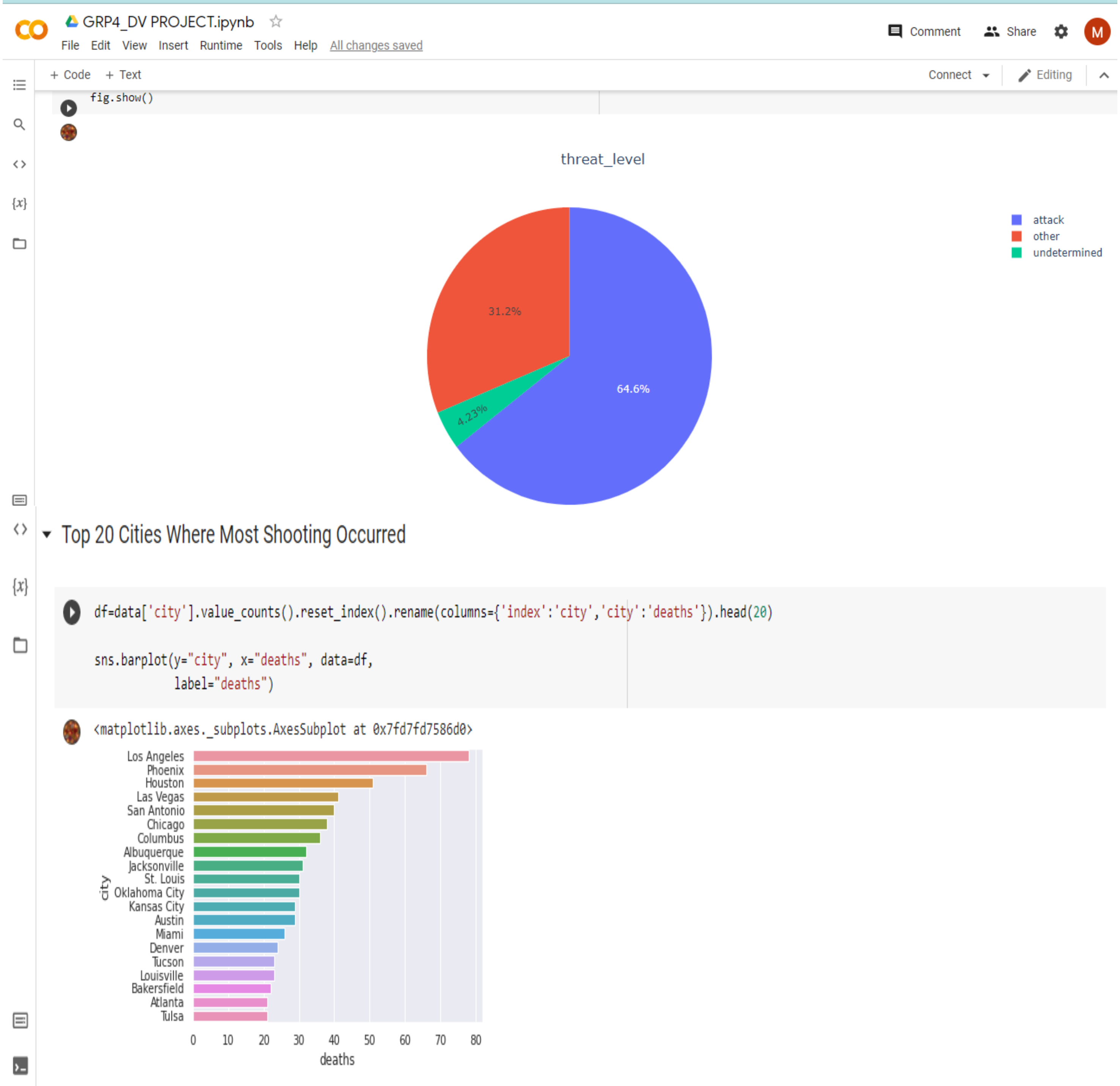


```
fig.update_layout(title="Signs_of_mental_illness",title_x=0.5)
fig.show()
```



Threat Level

```
df=data['threat_level'].value_counts().reset_index().rename(columns={'index':'threat_level','threat_level':'count'})
fig = go.Figure([go.Pie(labels=df['threat_level'],values=df['count'])])

#fig.update_traces(hoverinfo='label+percent', textinfo='value+percent', textfont_size=15,insidetextorientation='radial')

fig.update_layout(title="threat_level",title_x=0.5)
fig.show()
```



```
fig.show()
```

Top 20 Cities Where Most Shooting Occurred

```
df=data['city'].value_counts().reset_index().rename(columns={'index':'city','city':'deaths'}).head(20)

sns.barplot(y="city", x="deaths", data=df,
            label="deaths")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fd7fd7586d0>
```

CONCLUSION

The burden of fatal police violence is an urgent public health crisis in USA. Mounting evidence shows that deaths at the hands of police disproportionately impact people of certain races and ethnicities, pointing to systemic racism in policing. Recent high – profile killings by police in the USA have prompted calls for more extensive and public data reporting on police violence. we gather information from some websites ,and the data may not be accurate .The above displayed information might get differ with other web sites and also the data may be varying.

REFERENCES

1. https://matplotlib.org/
2. https://towardsdatascience.com/seaborn
3. https://seaborn.pydata.org/
4. https://pandas.pydata.org

IPYNB FILE:

https://colab.research.google.com/drive/1tKRCbHO1a2kEMHmgrAvuwDnKizJjJb0s?usp=sharing

Thank you