Visual Communication & Information Design

CMT212

Student Number: 1766249

Date Set: 5th March 2018

E-Submission Date: 9:30 am 8th May 2018

Contents

Introduction

For the Visual Communication and Information Design module (CMT212) final assignment we were asked to carry out an analysis of more than one datasets and present our findings in the form of visualizations and a report along with an analysis.

Moreover, there was no restrictions on what dataset we should work on, as well as on how to analyze and visualize those datasets. Therefore, to analyze my data I decided to use Python, as I believe is better suited for data analysis and statistical methods.

In addition, for the visualization part I have used Tableau and Python. Using python's libraries like matplolib and pandas you cannot create advance visualizations so I have used those to summarize and make my datasets easier to read and understand. The tableau software was used for the more detailed and advanced visualizations which Python language alone could not create.

Datasets Summary

The datasets I have chosen were based on the U.S police deaths from 1791 until 2016, the police killings during 2015 and the murders committed in the U.S in 2015.

The murders committed in the U.S dataset was not used because it was not of the same theme as the other two as well as lacked a lot of import information. Furthermore, the amount of data from the other two datasets was enough to create multiple visualizations.

The primary source of the police deaths data is the Officer Down Memorial Page, started in 1996 by a college student who is now a police officer and who continues to maintain the database (dmil, n.d.). The data consists of the person name, police department, year, date, cause of death etc. Overall it consists all the basic information of the deceased.

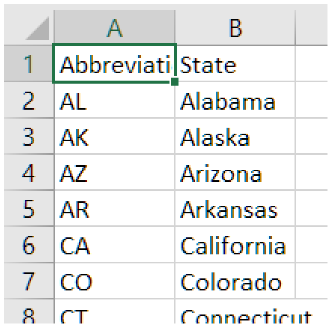
Additionally, the police killings dataset contains the data behind the story Where Police Have Killed Americans in 2015 and it's based on the Guardian's database on police killings to census data from the American Community Survey (dmil, 2015). It also consists of the tract population of state, ethnicity and income of the person. The primary purpose of census tracts is to provide a stable set of geographic units for the presentation of statistical data. Census tracts generally have a population size between 1,200 and 8,000 people, with an optimum size of 4,000 people.

Data analysis and Visualization

First, taking into consideration all the variables and the amount of information that both datasets had I decided to start with the Police Killings dataset. To make the dataset easier to read and work with on Python and tableau mainly, I had to change the state column from abbreviations to the full name.

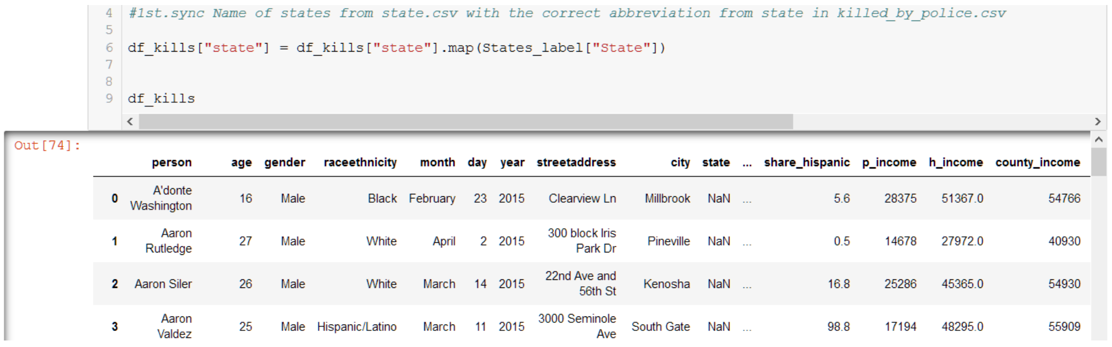
Therefore, a csv file consists of two columns state names and abbreviations was downloaded (Figure 1.0) (jasonong, n.d.). Using Python 3 language on Jupyter Notebook the states file was going to be used to map it with the police killings 2015 csv, in order to replace the state abbreviations appropriately with their name.

Unfortunately, there were some problems that were to solve. First a NaN value was replacing the state abbreviations whenever the map () was used (Figure 1.1).



	A	B
1	Abbreviati	State
2	AL	Alabama
3	AK	Alaska
4	AZ	Arizona
5	AR	Arkansas
6	CA	California
7	CO	Colorado
8	CT	Connecticut

Figure 1.0



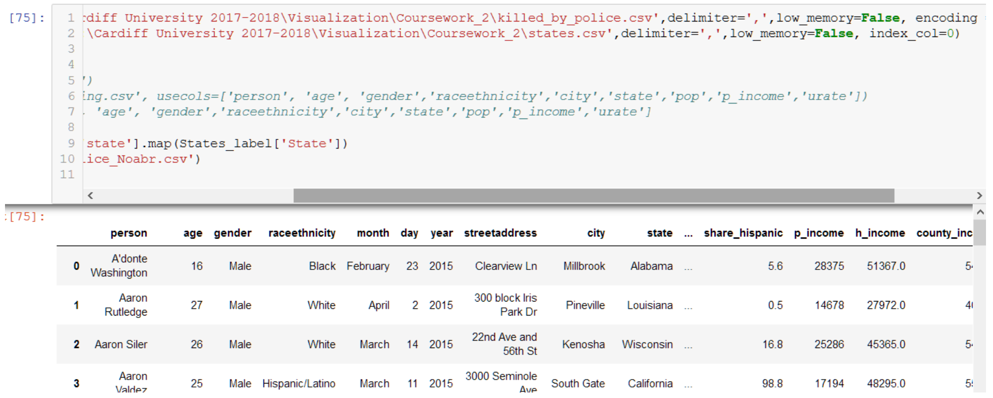
```
#1st.sync Name of states from state.csv with the correct abbreviation from state in killed_by_police.csv

df_kills["state"] = df_kills["state"].map(States_label["State"])


df_kills
```

Out[74]:

	person	age	gender	raceethnicity	month	day	year	streetaddress	city	state	...	share_hispanic	p_income	h_income	county_income
0	A'donte Washington	16	Male	Black	February	23	2015	Clearview Ln	Millbrook	NaN	...	5.6	28375	51367.0	54766
1	Aaron Rutledge	27	Male	White	April	2	2015	300 block Iris Park Dr	Pineville	NaN	...	0.5	14678	27972.0	40930
2	Aaron Siler	26	Male	White	March	14	2015	22nd Ave and 56th St	Kenosha	NaN	...	16.8	25286	45365.0	54930
3	Aaron Valdez	25	Male	Hispanic/Latino	March	11	2015	3000 Seminole Ave	South Gate	NaN	...	98.8	17194	48295.0	55909

Figure 1.1

The reasons why it was returning NaN values was either because there were white spaces in one of the csv files, or the state column from the states.csv needed to be set to index. Therefore, I tried using skipinitialspace = True so that when mapping the two csv it would ignore any spaces, but it did not work. Also, using the replace () function was a bad idea, as it just literally replaced the first 52 rows of the dataset with the 52 U.S states. Therefore, I tried to set the State column to index, but that did not change anything, so I set the index column to 0 instead when reading the state.csv file instead,(Figure 1.2) and it worked.



```
[75]:
:diff University 2017-2018\Visualization\Coursework_2\killed_by_police.csv',delimiter=',',low_memory=False, encoding
\Cardiff University 2017-2018\Visualization\Coursework_2\states.csv',delimiter=',',low_memory=False, index_col=0)


')
ing.csv', usecols=['person', 'age', 'gender','raceethnicity','city','state','pop','p_income','urate'])
 'age', 'gender','raceethnicity','city','state','pop','p_income','urate']

state'].map(States_label['State'])
.ice_Noabr.csv')
```

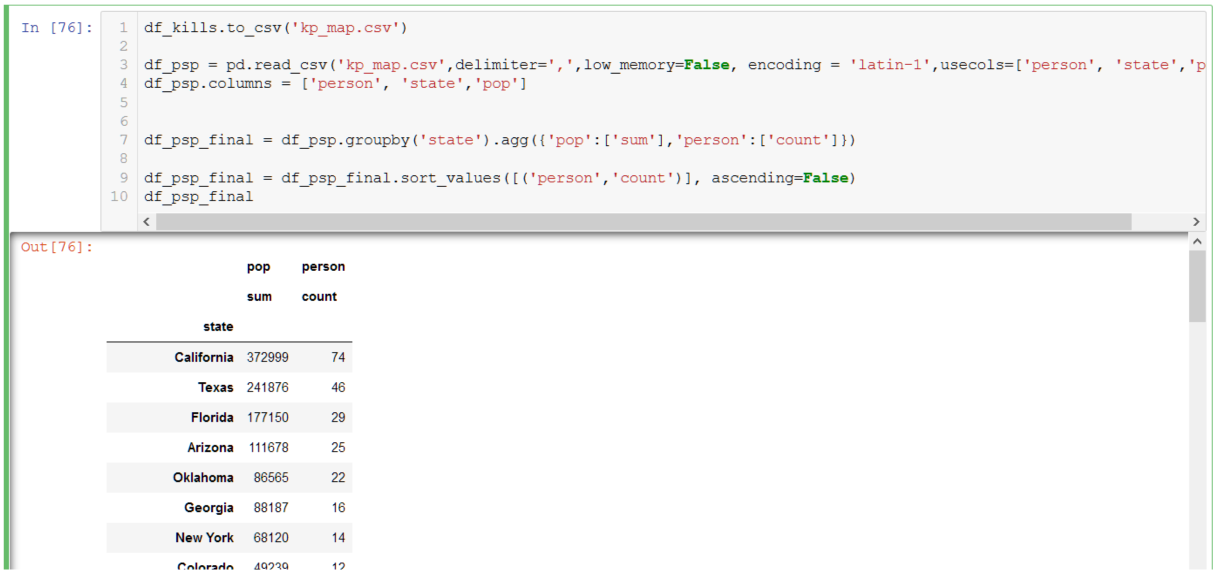
```
:[75]:
             person  age  gender  raceethnicity     month  day  year  streetaddress         city      state  ...  share_hispanic  p_income  h_income  county_inc
0  A'donte            16    Male          Black  February   23  2015  Clearview Ln      Millbrook    Alabama  ...             5.6     28375   51367.0  5
   Washington
1  Aaron              27    Male          White     April    2  2015  300 block Iris    Pineville  Louisiana  ...             0.5     14678   27972.0  4
   Rutledge                                                            Park Dr
2  Aaron Siler        26    Male          White     March   14  2015  22nd Ave and        Kenosha  Wisconsin  ...            16.8     25286   45365.0  5
                                                                       56th St
3  Aaron              25    Male Hispanic/Latino    March   11  2015  3000 Seminole    South Gate California  ...            98.8     17194   48295.0  5
   Valdez                                                              Ave
```

Figure 1.2

Furthermore, the first test I had decided to do was to check if the police killings (person) were increasing based on the U.S states population ratio (pop). So, my hypothesis was that the bigger the population ratio of a state, would mean the more the police killings.

To demonstrate and prove this theory a correlation test between the variables pop (tract population) and the person (police killings) to test their relationship status. If the two variables had a strong positive correlation (close to +1) therefore prove that as population increases, police killings increase as well.

Since the person column consists of names we cannot calculate the correlation between pop and person. Also, the pop column has a different tract population for each person. To solve this issues, we need to calculate using Python the population



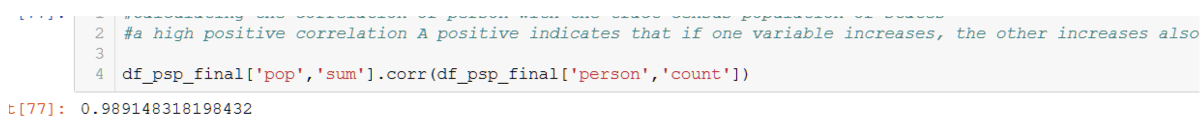
```
In [76]:
df_kills.to_csv('kp_map.csv')

df_psp = pd.read_csv('kp_map.csv',delimiter=',',low_memory=False, encoding = 'latin-1',usecols=['person', 'state','p
df_psp.columns = ['person', 'state','pop']


df_psp_final = df_psp.groupby('state').agg({'pop':['sum'],'person':['count']})

df_psp_final = df_psp_final.sort_values([('person','count')], ascending=False)
df_psp_final
```

```
Out[76]:
               pop  person
               sum   count
      state
 California 372999      74
      Texas 241876      46
    Florida 177150      29
    Arizona 111678      25
   Oklahoma  86565      22
    Georgia  88187      16
   New York  68120      14
   Colorado  49239      12
```

Figure 1.3

Sum for each state and count how many person killed in each state as shown in Figure 1.3.

Now that both columns were numbers we could use Python’s corr () function to calculate the correlation between the two Figure (1.4). A 0.98 correlation is a

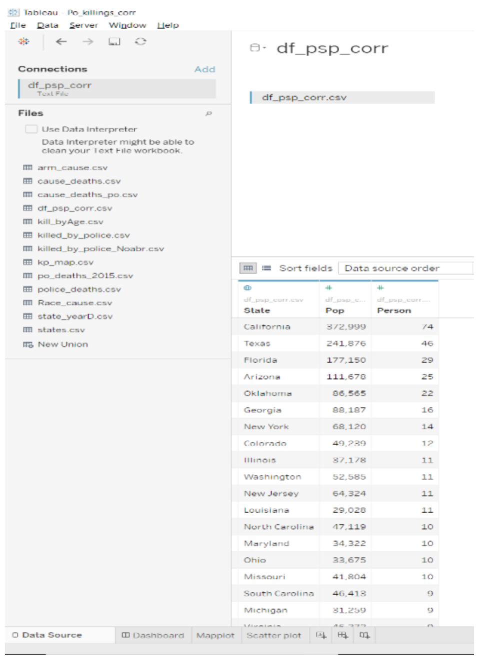


```
#calculating the correlation of person with the state census population of states
#a high positive correlation A positive indicates that if one variable increases, the other increases also

df_psp_final['pop','sum'].corr(df_psp_final['person','count'])
```

```
t[77]: 0.989148318198432
```

Figure 1.4

strong correlation which means the higher the population the more the police killings in each state. This can also be shown by creating a scatter and map plot visualization in tableau. To do that, we create a new pop vs person table as df_psp_corr.csv file. Then import it in tableau (Figure 1.5).

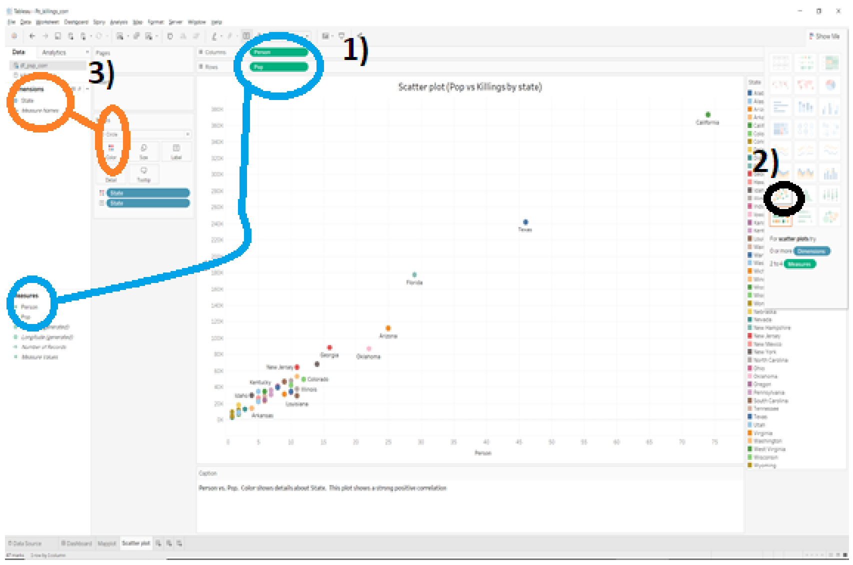


State	Pop	Person
California	372,999	74
Texas	241,876	46
Florida	177,150	29
Arizona	111,678	25
Oklahoma	86,565	22
Georgia	88,187	16
New York	68,120	14
Colorado	49,239	12
Illinois	37,178	11
Washington	52,585	11
New Jersey	64,324	11
Louisiana	29,028	11
North Carolina	47,119	10
Maryland	34,322	10
Ohio	33,675	10
Missouri	41,804	10
South Carolina	46,413	9
Michigan	31,259	9

Figure 1.5

Then by dragging the pop column to the rows tab and person column to the columns tab at the top of the tableau window (Figure 1.6) (1) and (2) choose the scatter plot design at the top right of the window to create a simple scatter plot. Also (3) to color and label the circles on the plot you should drag the state variable from the dimensions at the left side of the window and place it on top of the color in

the "Marks" section.

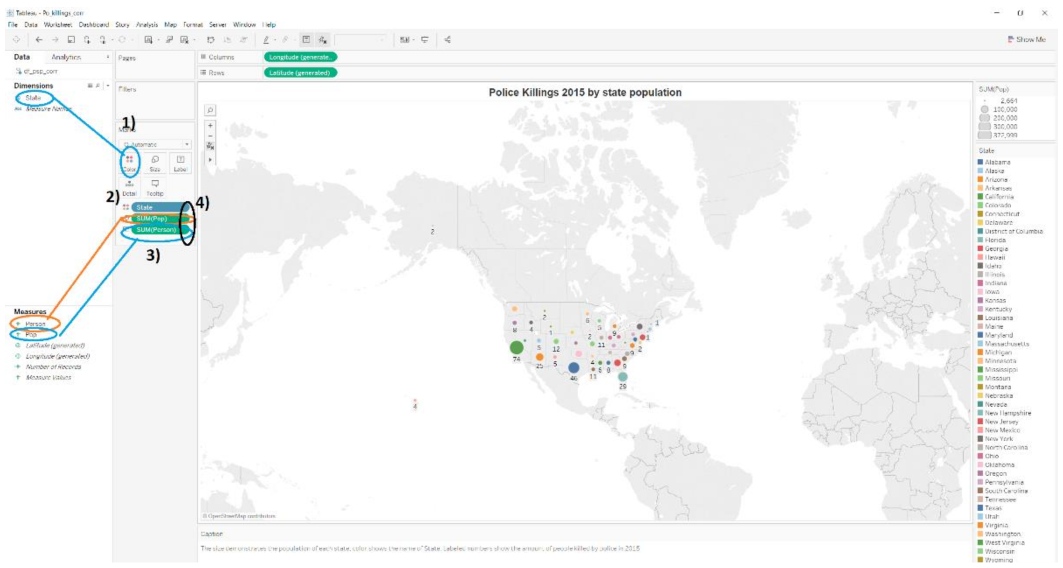


Caption

Person vs. Pop. Color shows details about State. This plot shows a strong positive correlation

Figure 1.6
Scatter plot also proves that there is a high correlation (as pop increases so does the person{killings})

Moreover, creating the map plot, was the same process although this time first the states column had to go at the color in the Marks section (Figure 1.7). Although this time we also had to change our variables measure to sum and count, for pop and person appropriately (4).


1)
2)
3)
4)
Police Killings 2015 by state population
State
SUM(Pop)
SUM(Person)
Measures
Person
Pop
© OpenStreetMap contributors

Caption

The size demonstrates the population of each state, color shows the name of State. Labeled numbers show the amount of people killed by police in 2015

Figure 1.7

Furthermore, to make those two visualizations more attractive and presentable we create a dashboard in tableau and add both (Figure 1.8).

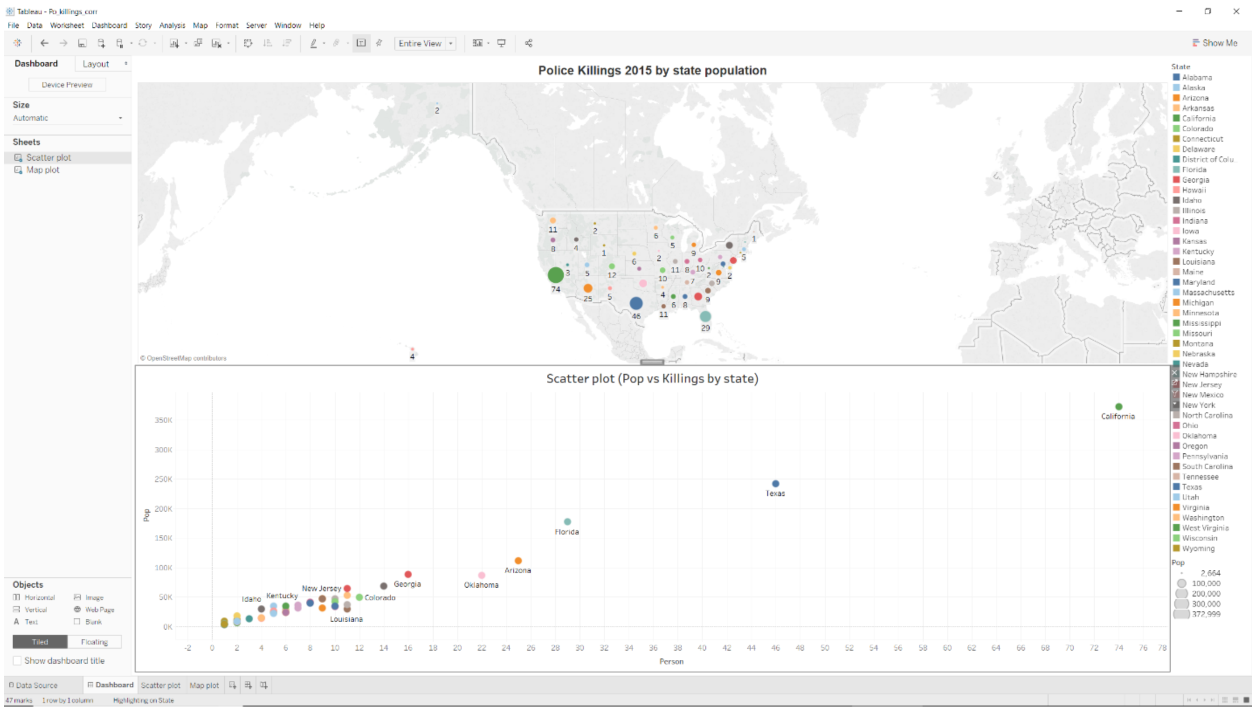


Figure 1.8

Another big factor of Tableau is that it allows you to filter all visualizations in the dashboard and sync them. In this case clicking one of the dots or states on the legend will show that state on both visualizations (Figure 1.9).

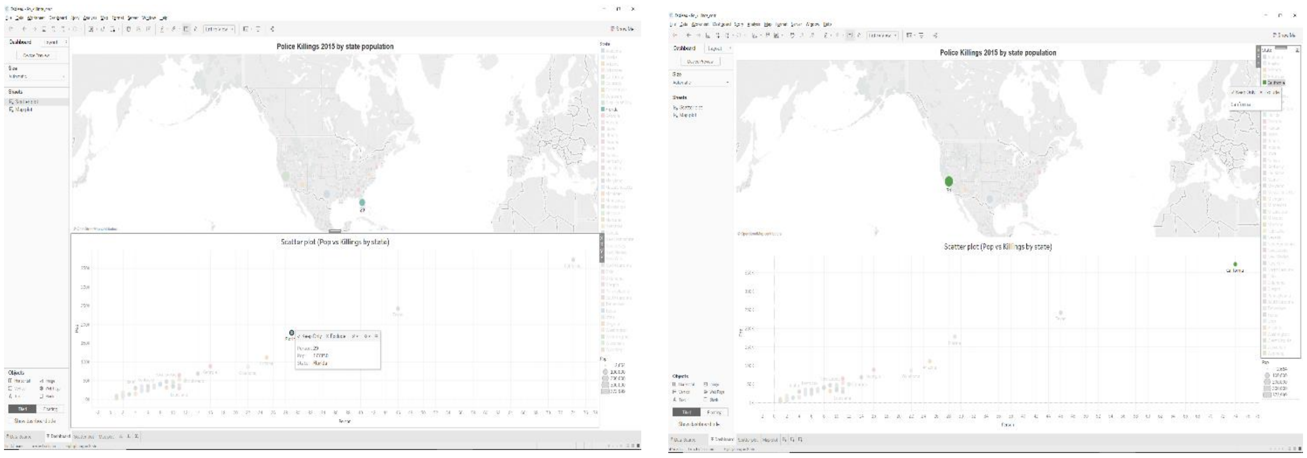
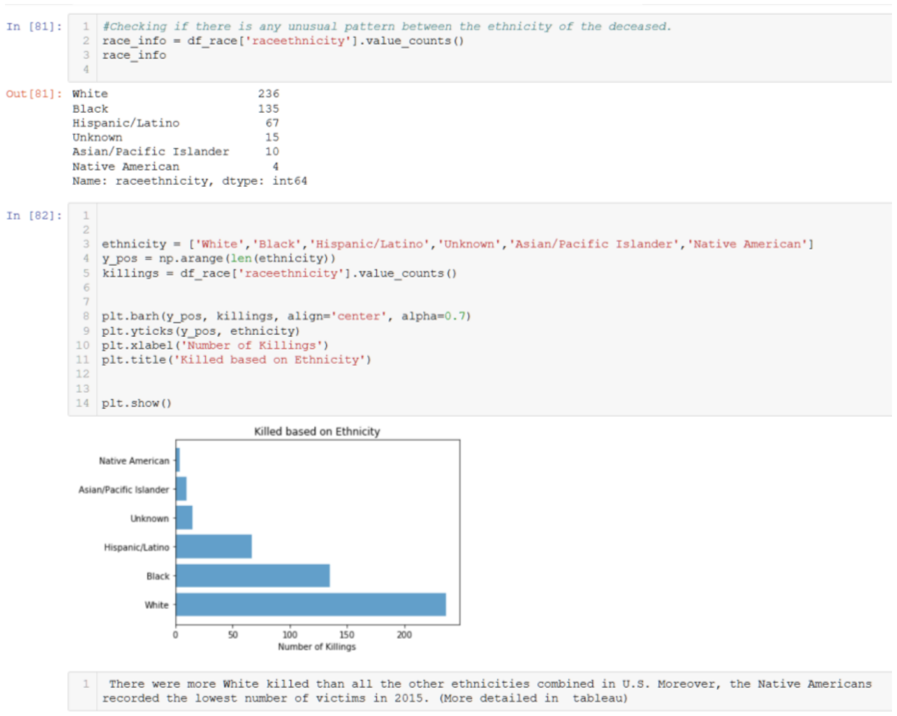

Police Killings 2015 by state population
Scatter plot (Pop vs Killings by state)

Figure 1.9

The next four tests and visualizations were used to check the data for any abnormalities or possible patterns. So, the first test attempted was police killings based on Ethnicity. To do that we create a new variable that counts the values of each Ethnicity under the variable race ethnicity and then create a bar chart using Python's matplotlib to demonstrate them (Figure 2.0).



```
In [81]:  #Checking if there is any unusual pattern between the ethnicity of the deceased.
          race_info = df_race['raceethnicity'].value_counts()
          race_info

Out[81]:  White                        236
          Black                        135
          Hispanic/Latino               67
          Unknown                       15
          Asian/Pacific Islander        10
          Native American                4
          Name: raceethnicity, dtype: int64
```

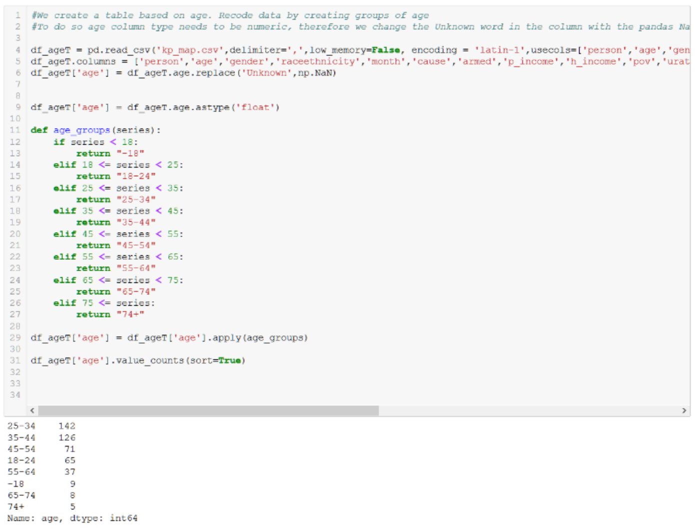
```
In [82]:

          ethnicity = ['White','Black','Hispanic/Latino','Unknown','Asian/Pacific Islander','Native American']
          y_pos = np.arange(len(ethnicity))
          killings = df_race['raceethnicity'].value_counts()


          plt.barh(y_pos, killings, align='center', alpha=0.7)
          plt.yticks(y_pos, ethnicity)
          plt.xlabel('Number of Killings')
          plt.title('Killed based on Ethnicity')


          plt.show()
```

```
 There were more White killed than all the other ethnicities combined in U.S. Moreover, the Native Americans
recorded the lowest number of victims in 2015. (More detailed in  tableau)
```

Figure 2.0

We can see from these bar chart that there were more white person police killings than all the other ethnicities combined in U.S.

Furthermore, an obvious analysis was the police killings by age. First, a data recoding process had to be done to create groups of age, to make the analysis easier and the visualization more accurate and easy on the eyes. To recode our data we replace all 'Unknown' values inside the age column with pandas NaN, since the pandas library recognizes NaN values as numeric as well and then the data type of our 'age' variable as float so that Python allows us to make alterations/renaming the 'age' variables elements (Figure 2.1).



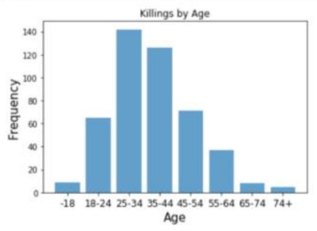
```
#We create a table based on age. Recode data by creating groups of age
#To do so age column type needs to be numeric, therefore we change the Unknown word in the column with the pandas Na

df_ageT = pd.read_csv('kp_map.csv',delimiter=',',low_memory=False, encoding = 'latin-1',usecols=['person','age','gen
df_ageT.columns = ['person','age','gender','raceethnicity','month','cause','armed','p_income','h_income','pov','urat
df_ageT['age'] = df_ageT.age.replace('Unknown',np.NaN)


df_ageT['age'] = df_ageT.age.astype('float')

def age_groups(series):
    if series < 18:
        return "-18"
    elif 18 <= series < 25:
        return "18-24"
    elif 25 <= series < 35:
        return "25-34"
    elif 35 <= series < 45:
        return "35-44"
    elif 45 <= series < 55:
        return "45-54"
    elif 55 <= series < 65:
        return "55-64"
    elif 65 <= series < 75:
        return "65-74"
    elif 75 <= series:
        return "74+"

df_ageT['age'] = df_ageT['age'].apply(age_groups)

df_ageT['age'].value_counts(sort=True)
```

```
25-34    142
35-44    126
45-54     71
18-24     65
55-64     37
-18        9
65-74      8
74+        5
Name: age, dtype: int64
```

Figure 2.1

In addition to the recoding, a pandas bar chart was created again to show the count of persons killed by the new age groups (Figure 2.2).

```
In [93]:

    #Create a bar chart to demonstrate the deaths by age groups

    label = ['25-34','35-44','45-54','18-24','55-64','-18','65-74','74+']
    x = df_ageT['age'].value_counts(sort=True)
    # this is for plotting purpose
    index = label
    plt.bar(index,x , alpha = 0.7)
    plt.xlabel('Age', fontsize=15)
    plt.ylabel('Frequency', fontsize=15)
    plt.xticks(index, label, fontsize=12, rotation=0)
    plt.title('Killings by Age')

    plt.show()
```



```
1.The age was normally distributed and the age of 25-34 has the highest number of occurences.
2.The under 18 group (-18) had 9 people killed all by gunshot and 7/9 were armed with firearm
3.The small group of 74+ had 5 people killed 4/5 by gunshot and 1/5 struck by vehicle and 1/5 was armed with a
firearm
```

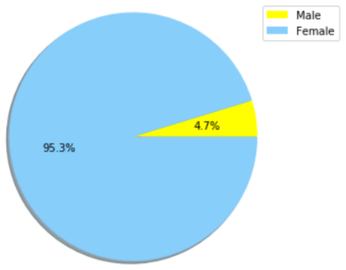
```
In [94]: df_ageT.to_csv('kill_byAge.csv')
```

Figure 2.2

Moreover, a simple pie chart was created again using the pandas library, to demonstrate police killings based on gender. By this pie chart we can see straight away that in 2015 the almost all the deceased were men (Figure 2.3).

Police killings based on gender

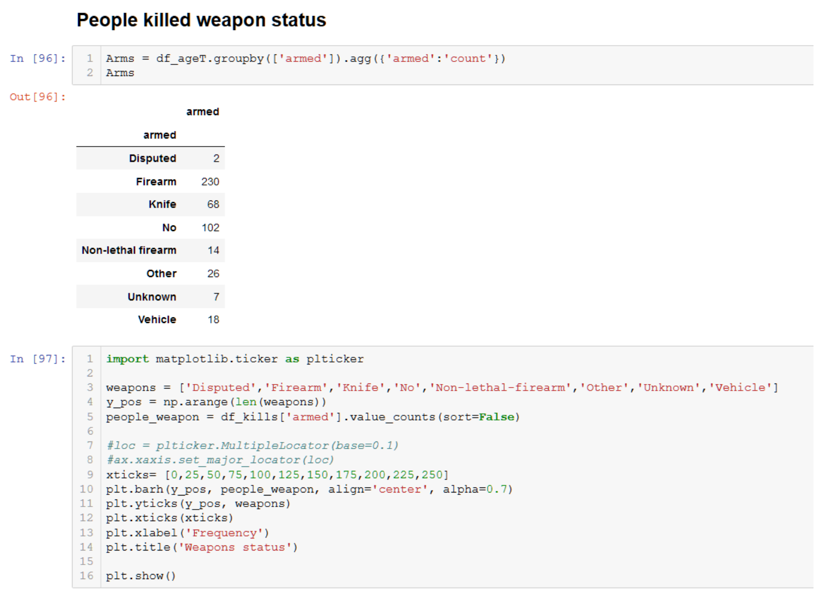
```
5]:
#Create pie chart
df_gender = df_kills.groupby('gender').agg({'gender':'count'})
labels = ['Male', 'Female']

Female =  df_gender['gender'].values[0]
Male =  df_gender['gender'].values[1]

sizes = [Female,Male]
colors = ['yellow', 'lightskyblue']
plt.pie(sizes, colors=colors, shadow=True, startangle=360,autopct='%1.1f%%')

plt.legend(labels, loc="best")
plt.axis('equal')
plt.tight_layout()
plt.show()
```



```
By observing the pie chart we can say that most of the victims of police killing in the U.S are male as
it consists of 95.3% of the total data recorded.
```

Figure 2.3

Next we analyze the people's killed weapon status/armed variable. Creating a variable to group police killings by the 'armed' variable and then use the aggregate [. agg ()] function to count how many of each value are (Figure 2.4) and then using pandas create a bar chart.



People killed weapon status

In [96]:
```
Arms = df_ageT.groupby(['armed']).agg({'armed':'count'})
Arms
```

Out[96]:

armed	armed
Disputed	2
Firearm	230
Knife	68
No	102
Non-lethal firearm	14
Other	26
Unknown	7
Vehicle	18

In [97]:
```
import matplotlib.ticker as plticker

weapons = ['Disputed','Firearm','Knife','No','Non-lethal-firearm','Other','Unknown','Vehicle']
y_pos = np.arange(len(weapons))
people_weapon = df_kills['armed'].value_counts(sort=False)

#loc = plticker.MultipleLocator(base=0.1)
#ax.xaxis.set_major_locator(loc)
xticks= [0,25,50,75,100,125,150,175,200,225,250]
plt.barh(y_pos, people_weapon, align='center', alpha=0.7)
plt.yticks(y_pos, weapons)
plt.xticks(xticks)
plt.xlabel('Frequency')
plt.title('Weapons status')

plt.show()
```

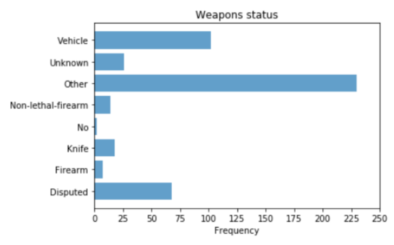
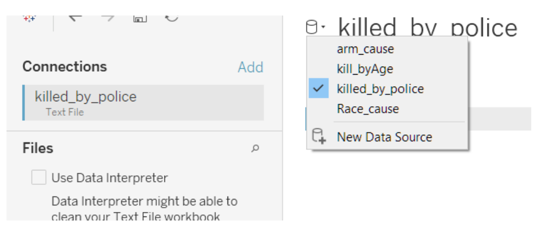


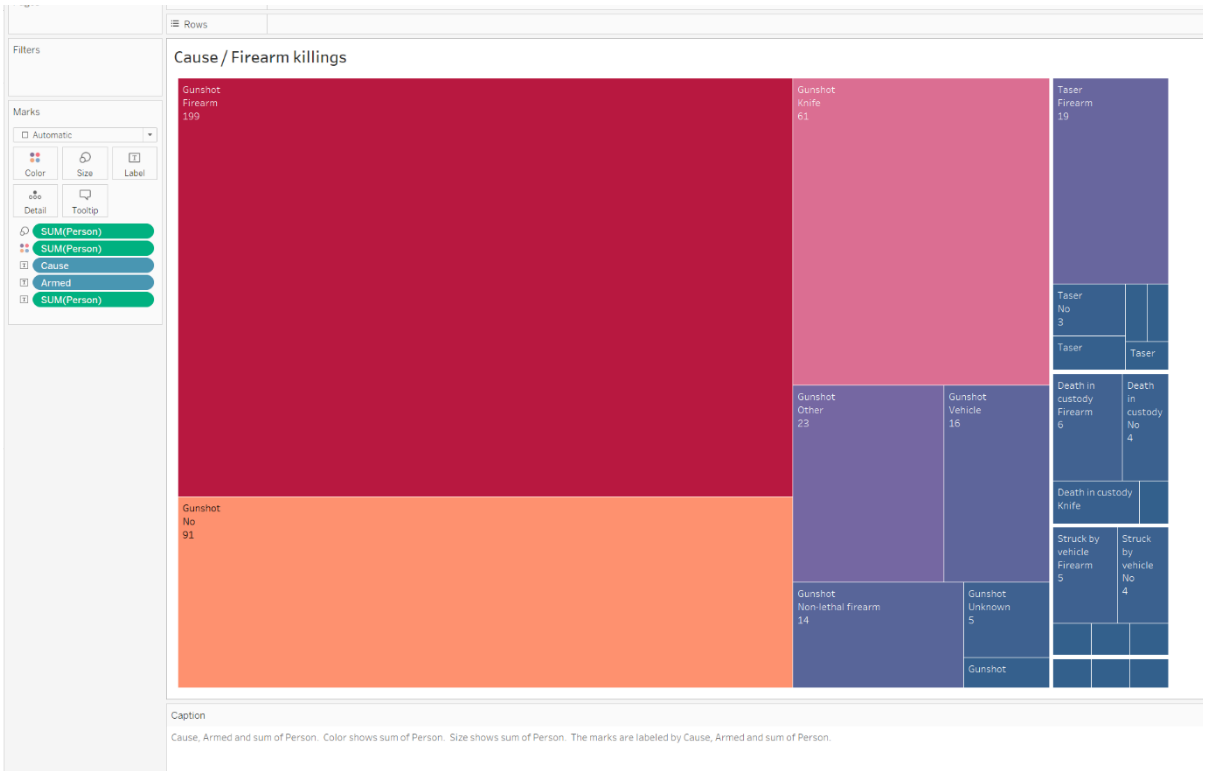
Figure 2.4

All four-previous visualization were created by Python's library pandas. Pandas does not give the same aesthetics or elegant designs and detail as tableau. Therefore, I have created a new csv file for each of those tests in order to visualize this data tests on Tableau.
Moving on, we import all four new csv files created using python on Tableau (Figure 2.5).

Figure 2.5



In Tableau we create one by one again all four visualizations. The Ethnicity, Age groups and Gender visualizations used the same approach as the scatter plot in Figure 1.6. For weapon status visualization a chart for tree maps was used. This is because the variable "cause" (cause of death) was used as well as 'armed' and sum of 'person' (police killings) (Figure 2.5).



Caption

Cause, Armed and sum of Person. Color shows sum of Person. Size shows sum of Person. The marks are labeled by Cause, Armed and sum of Person.

In my opinion the tree map chart that was used was the most appropriate chart in that case. This is because we want a quick, high level summary of the similarities and anomalies within multiple categories, "cause", "armed" and "person".

Afterwards, the same approach was used to create a dashboard with all four visualizations (Figure 2.6).

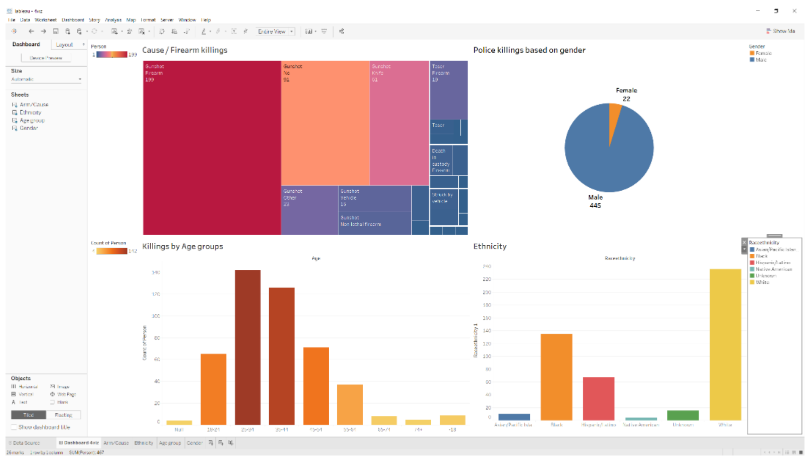
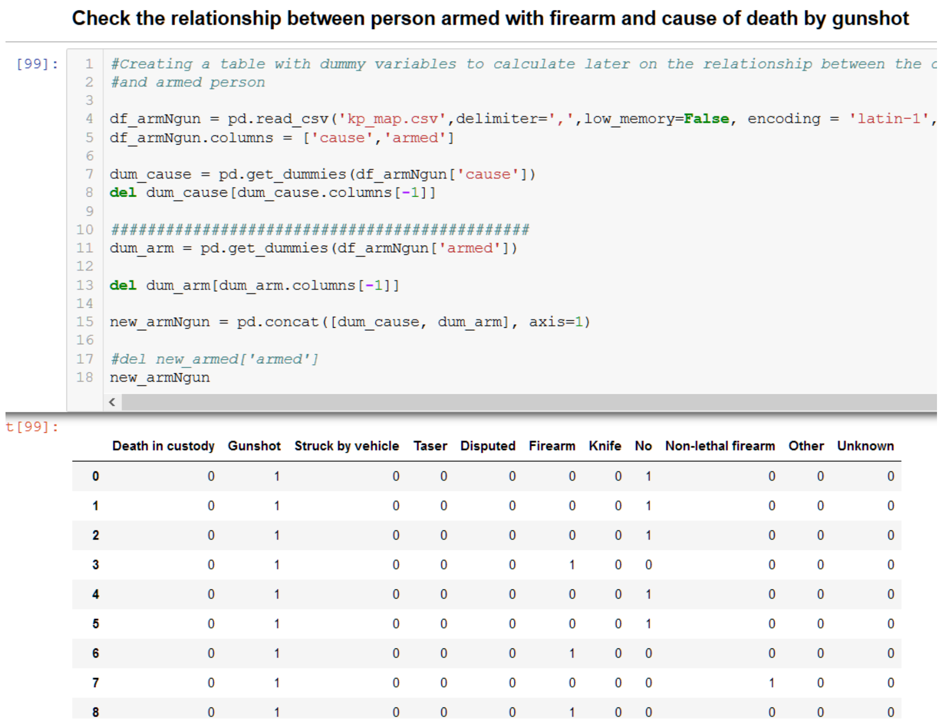


Figure 2.6

The final test for the police killings 2015 dataset was to create dummy variables to calculate the relationship between persons who were 'armed' with a firearm and their cause of death was by a gunshot. To calculate the relationship, we need to do a correlation test between the element 'gunshot' from variable "cause" and 'firearm' from 'armed' variable. Therefore, we create dummy variables this separates the two variables to sub-variables and give them the values 1 for True and 0 for False as shown Figure 2.7.



Check the relationship between person armed with firearm and cause of death by gunshot

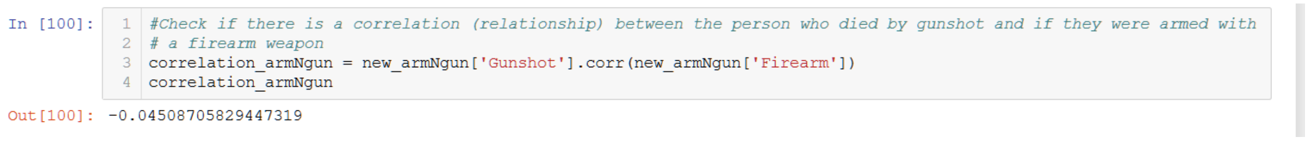
```
[99]:
#Creating a table with dummy variables to calculate later on the relationship between the c
#and armed person

df_armNgun = pd.read_csv('kp_map.csv',delimiter=',',low_memory=False, encoding = 'latin-1',
df_armNgun.columns = ['cause','armed']

dum_cause = pd.get_dummies(df_armNgun['cause'])
del dum_cause[dum_cause.columns[-1]]

###############################################
dum_arm = pd.get_dummies(df_armNgun['armed'])

del dum_arm[dum_arm.columns[-1]]

new_armNgun = pd.concat([dum_cause, dum_arm], axis=1)

#del new_armed['armed']
new_armNgun
```

t[99]:

	Death in custody	Gunshot	Struck by vehicle	Taser	Disputed	Firearm	Knife	No	Non-lethal firearm	Other	Unknown
0	0	1	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0	0
2	0	1	0	0	0	0	0	1	0	0	0
3	0	1	0	0	0	1	0	0	0	0	0
4	0	1	0	0	0	0	0	1	0	0	0
5	0	1	0	0	0	0	0	1	0	0	0
6	0	1	0	0	0	1	0	0	0	0	0
7	0	1	0	0	0	0	0	0	1	0	0
8	0	1	0	0	0	1	0	0	0	0	0

Figure 2.7

Now we can calculate the correlation between the two (Figure 2.8).



```
In [100]: #Check if there is a correlation (relationship) between the person who died by gunshot and if they were armed with
          # a firearm weapon
          correlation_armNgun = new_armNgun['Gunshot'].corr(new_armNgun['Firearm'])
          correlation_armNgun
Out[100]: -0.04508705829447319
```

Figure 2.8

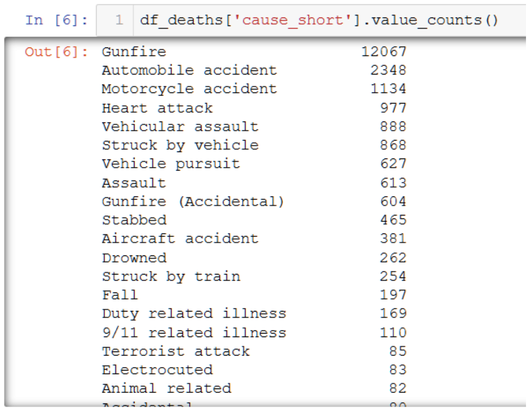
Based on the -0.045 correlation the person died by gunshot do not affect the person armed with a firearm, therefore there is no relationship between the two.

Data Analysis and Visualization Part B

As was previously stated, the second dataset is based on the police deaths from 1791 until 2016. This dataset consists of less variables. This made challenging to find any unique tests or visualizations to create.

The one obvious visualization that could be done with the police deaths dataset was to create a chart that would show the police deaths from 1971 until 2016. This could show us any unpredicted changes or events throughout the years.

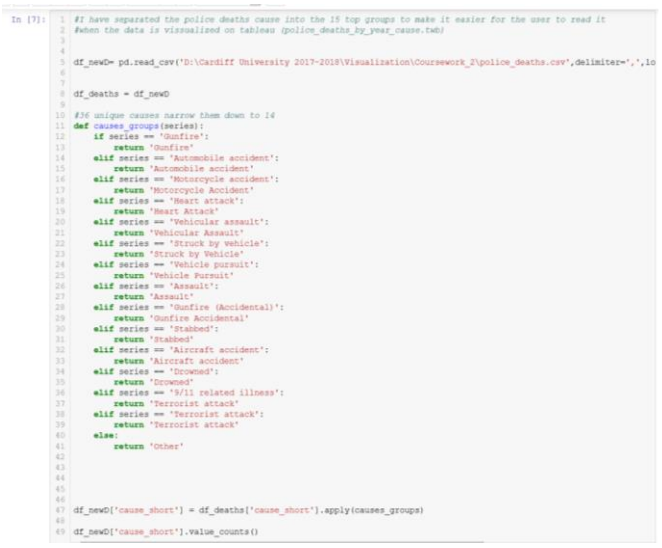
In the same way as the first dataset we import the police deaths dataset using Python on Jupyter notebook. To make the visualization more interesting we add the causes of the deceased as well, therefore we count how many elements are in “cause_short” variable and the values of each (Figure 3.0).



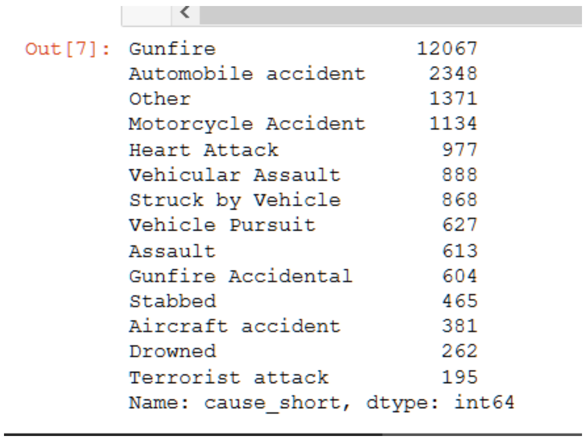
```
In [6]:  df_deaths['cause_short'].value_counts()

Out[6]:  Gunfire                      12067
         Automobile accident           2348
         Motorcycle accident           1134
         Heart attack                   977
         Vehicular assault              888
         Struck by vehicle              868
         Vehicle pursuit                627
         Assault                        613
         Gunfire (Accidental)           604
         Stabbed                        465
         Aircraft accident              381
         Drowned                        262
         Struck by train                254
         Fall                           197
         Duty related illness           169
         9/11 related illness           110
         Terrorist attack                85
         Electrocuted                    83
         Animal related                  82
```

Figure 3.0

Cause_short variable consists of many different elements this makes it difficult to analyze our data and it will overwhelm later our visualization with information. For this reason, we recode “cause short” into smaller groups. (Figure 3.1)



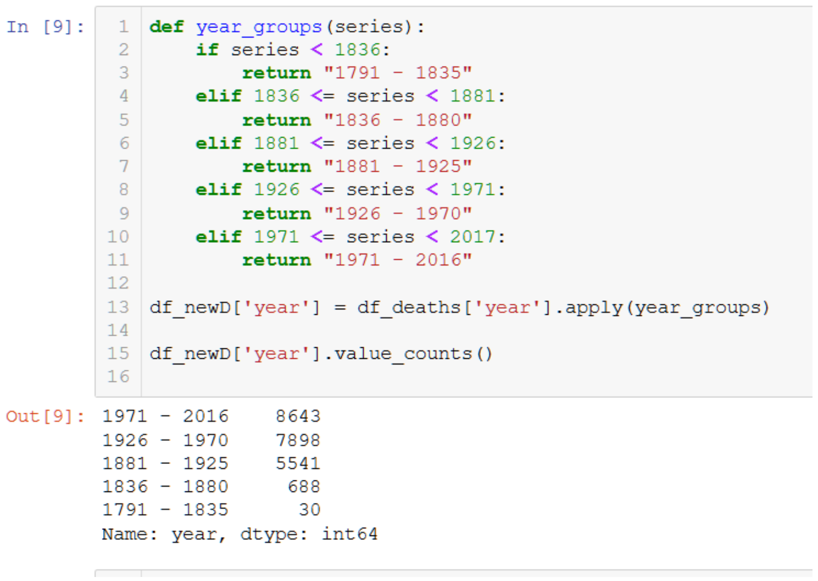
```
In [7]:
#I have separated the police deaths cause into the 15 top groups to make it easier for the user to read it
#when the data is vissualized on tableau (police_deaths_by_year_cause.twb)


df_newD= pd.read_csv('D:\Cardiff University 2017-2018\Visualization\Coursework_2\police_deaths.csv',delimiter=',',lo


df_deaths = df_newD

#36 unique causes narrow them down to 14
def causes_groups(series):
    if series == 'Gunfire':
        return 'Gunfire'
    elif series == 'Automobile accident':
        return 'Automobile accident'
    elif series == 'Motorcycle accident':
        return 'Motorcycle Accident'
    elif series == 'Heart attack':
        return 'Heart Attack'
    elif series == 'Vehicular assault':
        return 'Vehicular Assault'
    elif series == 'Struck by vehicle':
        return 'Struck by Vehicle'
    elif series == 'Vehicle pursuit':
        return 'Vehicle Pursuit'
    elif series == 'Assault':
        return 'Assault'
    elif series == 'Gunfire (Accidental)':
        return 'Gunfire Accidental'
    elif series == 'Stabbed':
        return 'Stabbed'
    elif series == 'Aircraft accident':
        return 'Aircraft accident'
    elif series == 'Drowned':
        return 'Drowned'
    elif series == '9/11 related illness':
        return 'Terrorist attack'
    elif series == 'Terrorist attack':
        return 'Terrorist attack'
    else:
        return 'Other'





df_newD['cause_short'] = df_deaths['cause_short'].apply(causes_groups)

df_newD['cause_short'].value_counts()
```



```
Out[7]: Gunfire                  12067
        Automobile accident       2348
        Other                     1371
        Motorcycle Accident       1134
        Heart Attack               977
        Vehicular Assault          888
        Struck by Vehicle          868
        Vehicle Pursuit            627
        Assault                    613
        Gunfire Accidental         604
        Stabbed                    465
        Aircraft accident          381
        Drowned                    262
        Terrorist attack           195
        Name: cause_short, dtype: int64
```

Figure 3.1

Then, we do the same for variable year (Figure 3.2).



```
In [9]:
def year_groups(series):
    if series < 1836:
        return "1791 - 1835"
    elif 1836 <= series < 1881:
        return "1836 - 1880"
    elif 1881 <= series < 1926:
        return "1881 - 1925"
    elif 1926 <= series < 1971:
        return "1926 - 1970"
    elif 1971 <= series < 2017:
        return "1971 - 2016"

df_newD['year'] = df_deaths['year'].apply(year_groups)

df_newD['year'].value_counts()

Out[9]: 1971 - 2016    8643
        1926 - 1970    7898
        1881 - 1925    5541
        1836 - 1880     688
        1791 - 1835      30
        Name: year, dtype: int64
```

Figure 3.2

After we have recoded and created our new variable groups, we code a new table that consists of those two new variable groups and output it as a new csv.

Furthermore, import the new csv called "cause_deaths.csv" to Tableau. Then create a new worksheet and drag year and person (right click ->measures count) [1] to columns and rows appropriately. Moreover, drag the "cause_short" variable to the color Mark section to show the cause of death of the police in colors as shown Figure 3.3 [2]. In addition, you can right-click on the right section of the window [4] and add filter and captions of your choice.

This area chart visualization shows clearly that 1930 is one of the deadliest year in law enforcement history and also, we can spot that on one specific day of 2001 [3] there is a big increase in deaths, and that is the 9/11 incident where 73 police officers died.

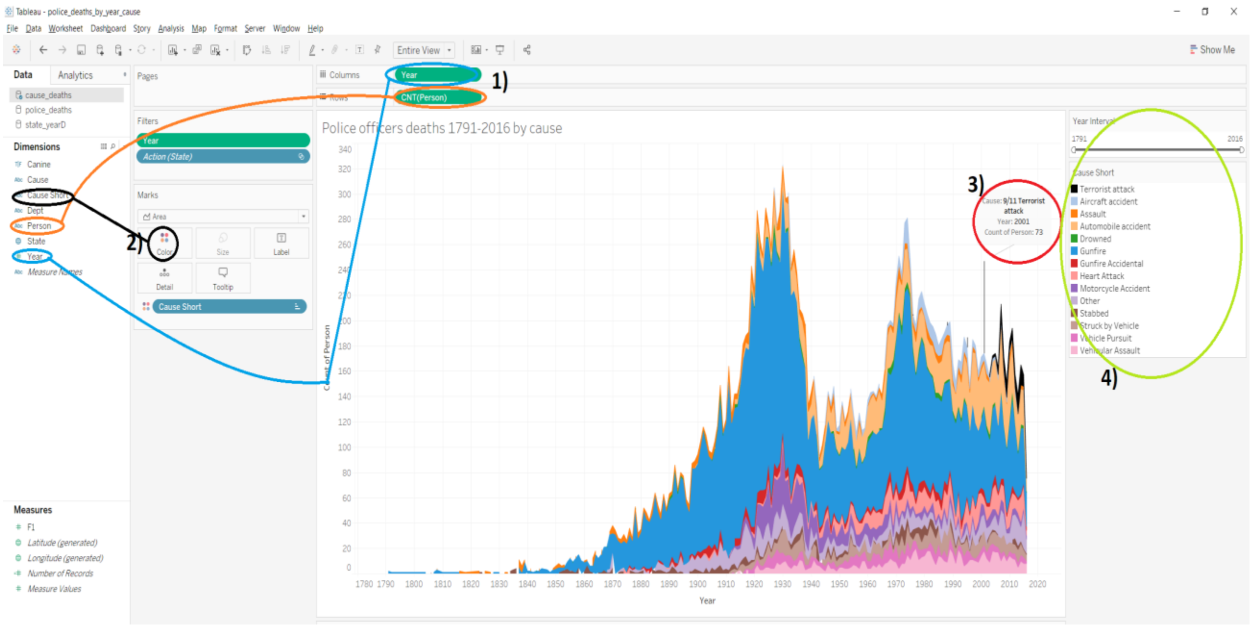


Figure 3.3

In addition, another two graphs were created one that identifies all canine (dog) officers, showing their increase in deaths by the years and another that count the police deaths by state and then placed into a dashboard Figure (3.4).

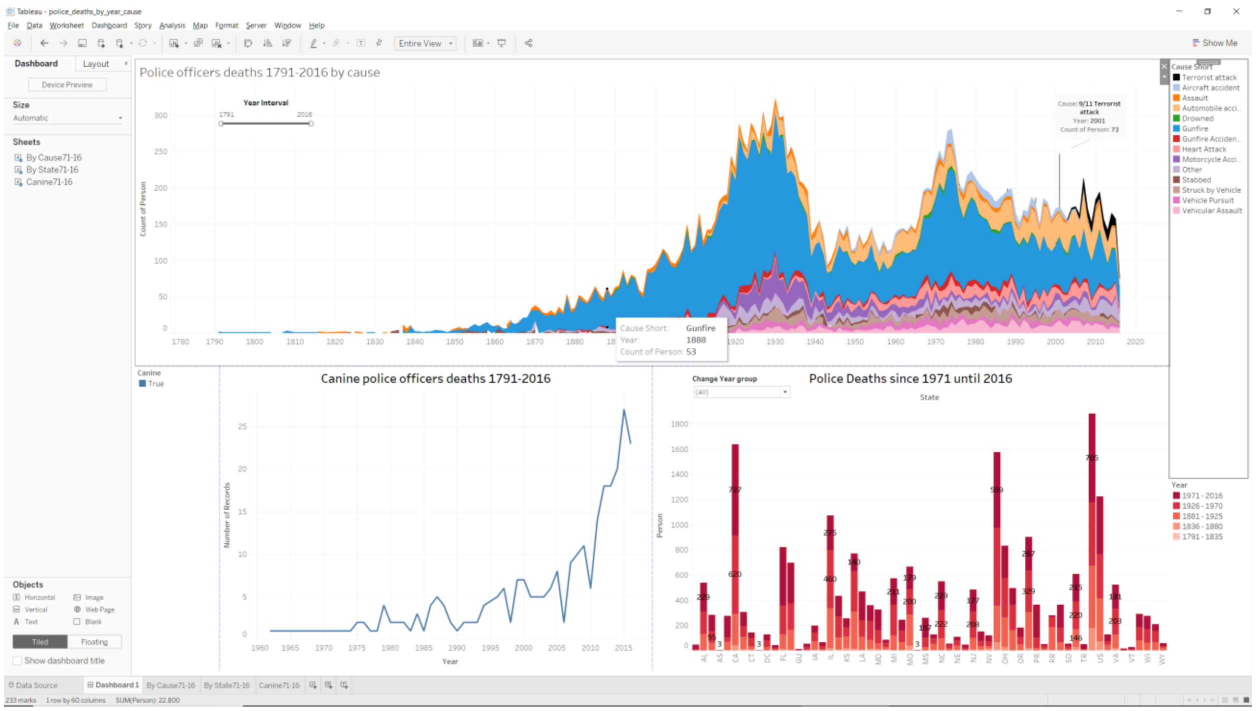


Figure 3.4

The next figure also shows how the user can interact with all three visualizations (Figure 3.5,3.6).

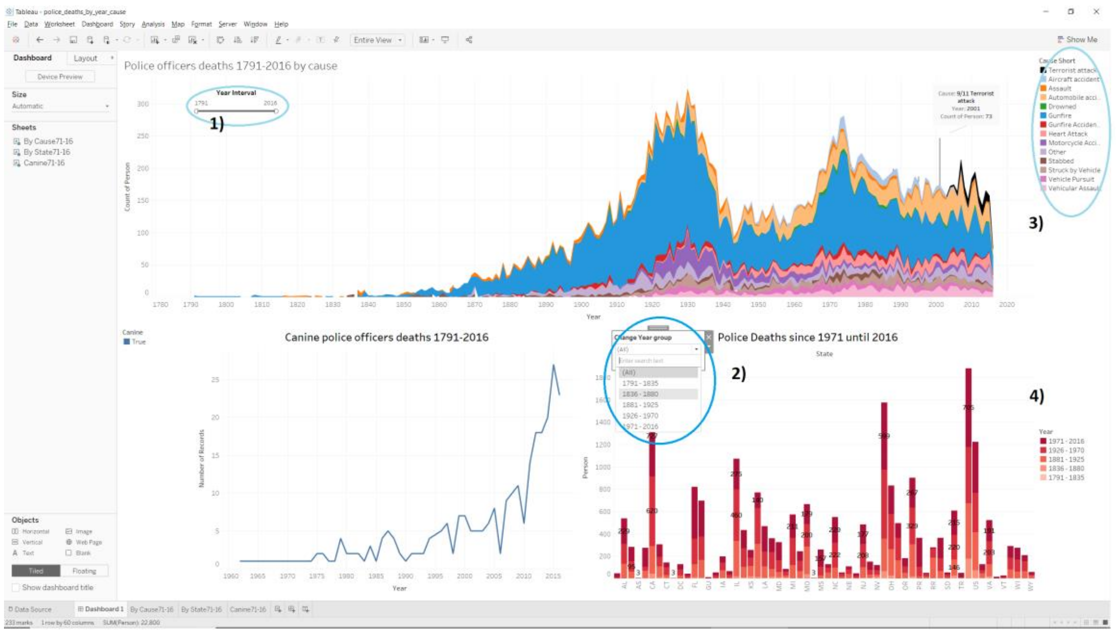

Police officers deaths 1791-2016 by cause
Year Interval
1)
2)
3)
4)
Canine police officers deaths 1791-2016
Change Year group
(All)
1791 - 1835
1836 - 1880
1881 - 1925
1926 - 1970
1971 - 2016
Police Deaths since 1971 until 2016

Figure 3.5
Adjust the year you want to see on the table

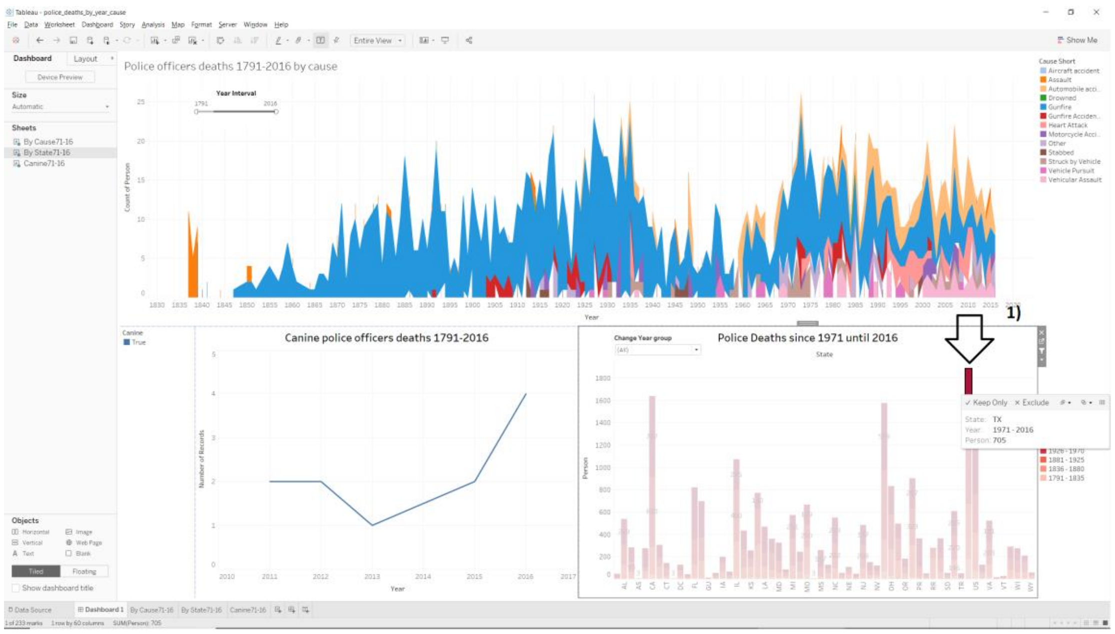
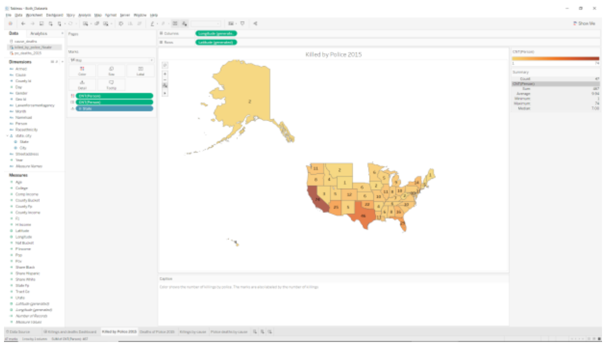


Figure 3.6
Click on a bar to show the appropriate year on all 3 visualizations

This was done by the filter option of tableau and by adjusting all three datasets of the visualization to have at least one key variable.

Combination of Datasets

The last visualization designed, to show the difference in numbers of police deaths and police killings in 2015 as well as their cause of death. These visualizations used both datasets. By importing both datasets we separately create each visualization as shown in Figures 4.0,4.1



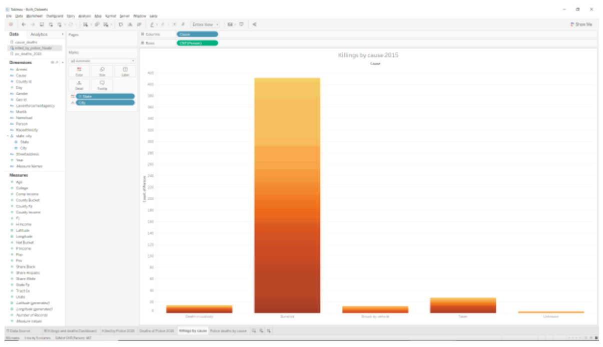


Figure 4.0

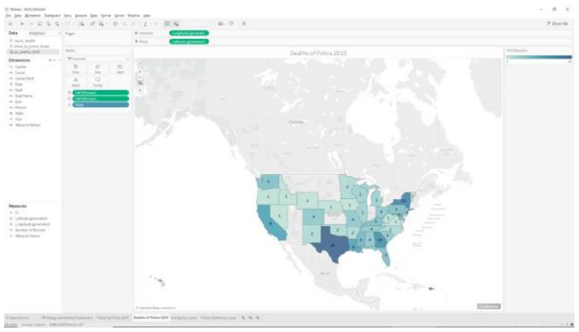
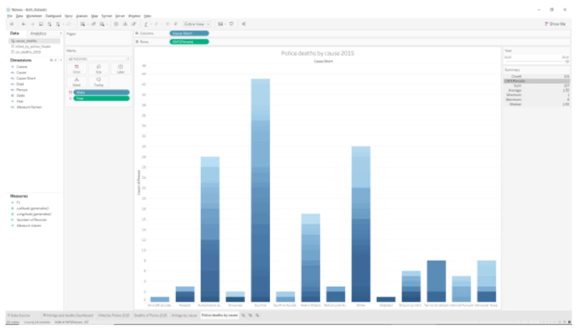



Figure 4.1

Then theses worksheets are added to a dashboard (Figure 4.2)

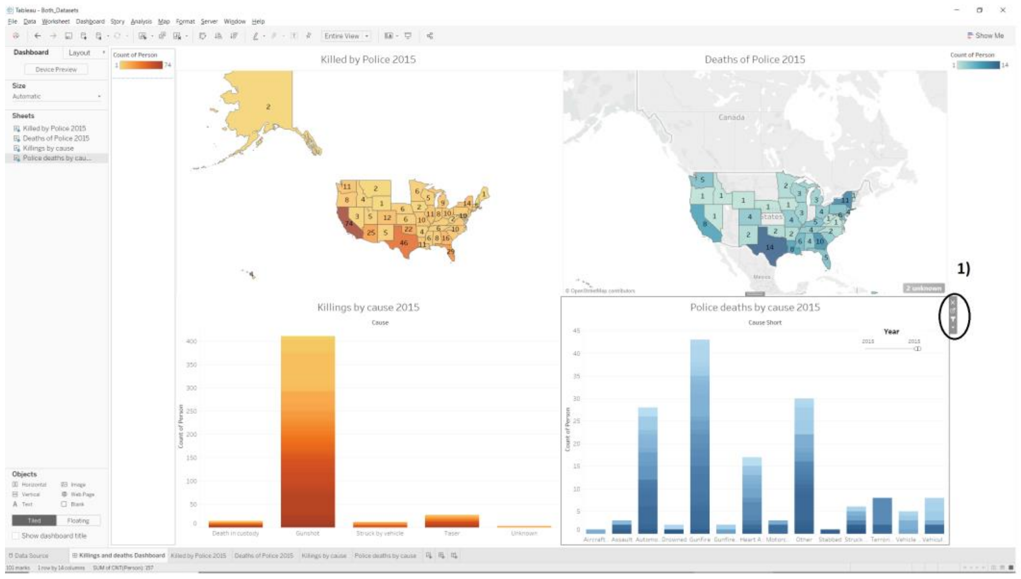
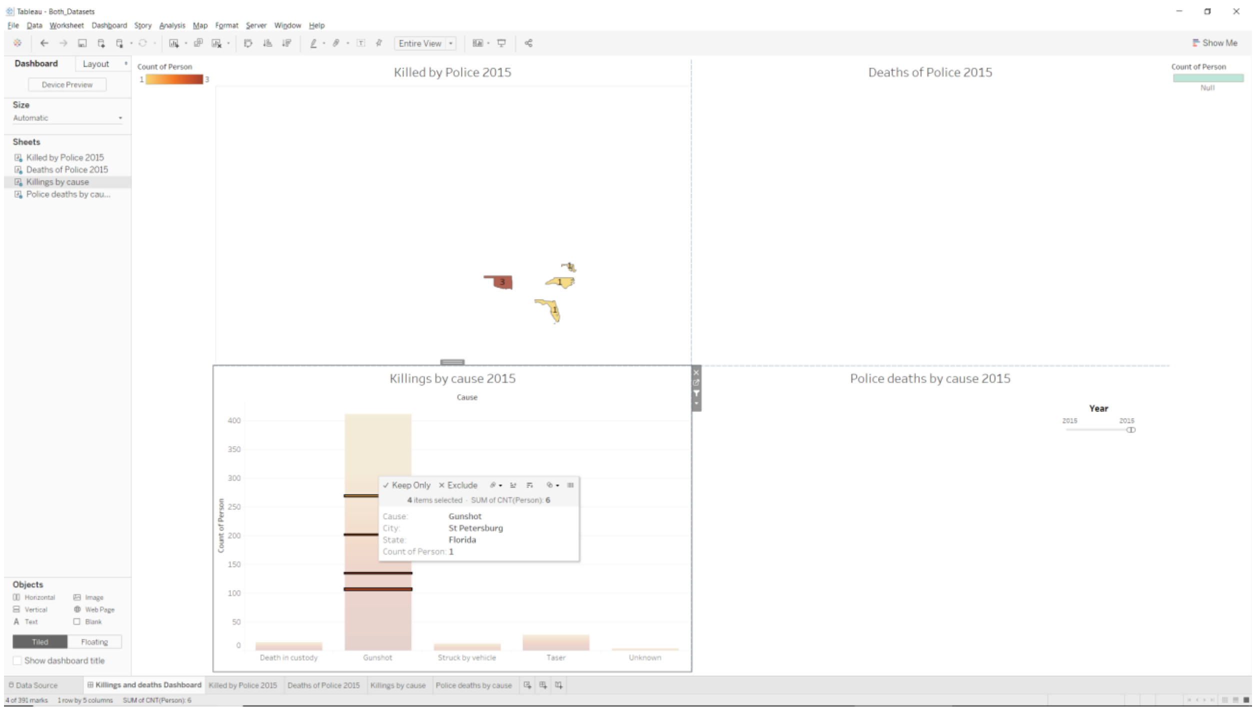


Figure 4.2

Again, all visualizations are interactive. Using the filter function of tableau, we can sync all four visualizations and allow the user to see the changes made to all four when it interacts with any of the visualizations.

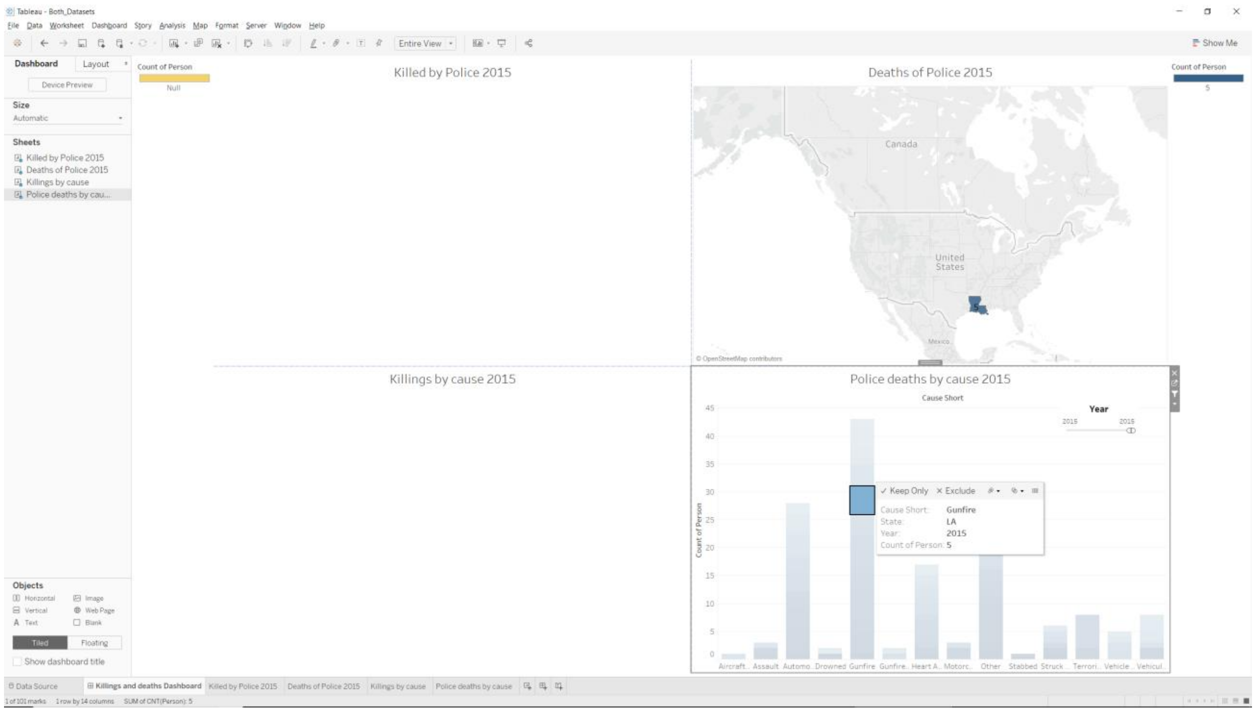


4 items selected · SUM of CNT(Person): 6

Cause: Gunshot
City: St Petersburg
State: Florida
Count of Person: 1

4 of 391 marks 1 row by 5 columns SUM of CNT(Person): 6

Figure 4.2
By pressing control + mouse click you can choose to highlight more than one value



Cause Short: Gunfire
State: LA
Year: 2015
Count of Person: 5

1 of 101 marks 1 row by 14 columns SUM of CNT(Person): 5

Figure 4.3

Critical View

This coursework was challenging in my opinion. For the first time, I had to find multiple datasets and carry out an analysis and present it in the form of a visualization. Based on the datasets I have chosen, I believe I have done a decent job analyzing and visualize them in a way that the user would understand the data itself and what it shows.

Both datasets might have the same "theme" however, they don't share any common information for me to be able to test their relationship. Therefore, should have chosen my datasets more carefully before I had submitted them. Two statistical tests were made that to show the relationship between variables but, more statistical tests should have been done to justify some of the visualizations. I wanted to run a clustering test between the two datasets but as I have said before not only I was not able to merge the two but most of the statistical tests I went through needed numerical data to run on python, whereas both datasets had more categorical.

Moreover, the visualizations I believe are on point, attractive and easy for the user to interact and understand them. Tableau is easy to work with and can do complex analysis as well as it is great for data exploration. I am glad I have decided to use Tableau for this project as it is expensive to buy, therefore it was a good opportunity for me to learn how to use it. I believe it really suit big companies because of its easy sharing platform, its online community and fast data engine. On the other hand, D3 could have provided me with a bigger library of visualization designs and an online (html) dashboard.

In the end, I have found this coursework a really good experience that will help me in the near future and I am looking forward working on similar projects.

References

dmil, 2015. *Github.* [Online]
Available at: https://github.com/fivethirtyeight/data/tree/master/police-killings
[Accessed 5 May 2018].

dmil, n.d. *Github.* [Online]
Available at: https://github.com/fivethirtyeight/data/tree/master/police-deaths
[Accessed 5 May 2018].

Hammid, O. A., 2017. *Towards Data Science.* [Online]
Available at: https://towardsdatascience.com/exploratory-data-analysis-visualizing-police-killings-in-the-u-s-in-2015-2cb5122c6cbe
[Accessed 12 April 2018].

jasonong, n.d. *Github.* [Online]
Available at: https://github.com/jasonong/List-of-US-States/blob/master/states.csv
[Accessed 14 Apri 2018].