Accidents and personal injuries in Colombia

The Accidents in Colombia project is an initiative to improve safety in the country. We will study the behavior of accidents and how they affect different ages and genders. In addition, we are going to use machine learning algorithms to identify patterns and trends in accidents, as well as to analyze the type of weapons involved. This information will help us create effective programs and policies to reduce accidents and improve safety across the country. This initiative is an excellent opportunity to contribute to Colombia's security and improve the lives of its citizens.

This project consist on a analysis of data of accidents in Colombia. The goal is to find patterns and factors in the incidence of accidents in country. The analysis is done with Policia Nacional compiled data, which include districts and cities as well as number of accidents, date, gender and behavior.

We will be use various types of statistical methods to iondentufy patterns and relations which are useful for the goberment and the public. This patterns and relaionsto be will use to make recomendation to improve the public policies to try to reduce the number of accidents in the country.

About dataset

In this dataset we have 1 million accidents from January 2010 to August 2022. their causes, weapon or means by which the event occurred. These data are from Policia Nacional and extracted by datos abiertos Colombia

What we want to figure out with this analysis?

- How many people per year, month and day have an accidents and personal injuries?
- Which departments and boroughs with the most accidents and personal injuries?
- Which weapons are the most used in personal injuries by gender and department?
- When ocurrs this accicents by month day and week?
- What gender is the most affected by the accidents?

import libraries

In [ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.api as sm
```

We loaded the dataset Personal Injuries and Traffic Accidents from the Policia Nacional

In [ ]:
```
df = pd.read_csv("C:/Users/Jorge/Downloads/Reporte_Lesiones_Personales_y_en_Accidente_
df
```

Out[ ]:

	DEPARTAMENTO	MUNICIPIO	CODIGO DANE	ARMAS MEDIOS	FECHA HECHO	GENERO	
0	ANTIOQUIA	GIRARDOTA	5308000	ARMA BLANCA / CORTOPUNZANTE	1/01/2010	FEMENINO	
1	ANTIOQUIA	GIRARDOTA	5308000	ARMA BLANCA / CORTOPUNZANTE	1/01/2010	MASCULINO	
2	ANTIOQUIA	MUTATÁ	5480000	ARMA BLANCA / CORTOPUNZANTE	1/01/2010	MASCULINO	
3	ANTIOQUIA	NECOCLÍ	5490000	ARMA BLANCA / CORTOPUNZANTE	1/01/2010	FEMENINO	
4	ATLÁNTICO	BARRANQUILLA (CT)	8001000	ARMA BLANCA / CORTOPUNZANTE	1/01/2010	FEMENINO	
...	...	...	...	...	...	...	
1047244	CESAR	VALLEDUPAR (CT)	20001000	VENENO	3/05/2022	MASCULINO	
1047245	HUILA	OPORAPA	41503000	VENENO	16/06/2022	FEMENINO	ADC
1047246	TOLIMA	IBAGUÉ (CT)	73001000	VENENO	17/04/2022	MASCULINO	
1047247	CUNDINAMARCA	COTA	25214000	SIN EMPLEO DE ARMAS	30/03/2022	MASCULINO	
1047248	CUNDINAMARCA	GUADUAS	25320000	SIN EMPLEO DE ARMAS	10/06/2022	MASCULINO	

1047249 rows × 9 columns

We start to understand the dataset

- Check the date
- Check the shape of the data
- Review the quality of the data, verify if are null values
- Review the format of the columns

In [ ]:
```
#revisar desde que fecha empieza y termina
print(df['FECHA HECHO'].min())
print(df['FECHA HECHO'].max())
```

```
1/01/2010
9/12/2021
```

In [ ]:
```
df.describe(include='object')
```

Out[ ]:

	DEPARTAMENTO	MUNICIPIO	CODIGO DANE	ARMAS MEDIOS	FECHA HECHO	GENERO	GRUPO ETARIO
count	1047249	1047249	1047249	1047249	1047249	1047249	1046285
unique	32	1023	1250	45	4626	5	5
top	CUNDINAMARCA	BOGOTÁ D.C. (CT)	11001000	CONTUNDENTES	1/01/2020	MASCULINO	ADULTOS
freq	134439	61226	61226	368472	1346	592363	853564

In [ ]:
```
round(df.describe())
```

Out[ ]:

	CANTIDAD
count	1047249.0
mean	2.0
std	2.0
min	1.0
25%	1.0
50%	1.0
75%	1.0
max	114.0

In [ ]:
```
df.shape
```

Out[ ]:
```
(1047249, 9)
```

In [ ]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1047249 entries, 0 to 1047248
Data columns (total 9 columns):
 #   Column               Non-Null Count    Dtype 
---  ------               --------------    ----- 
 0   DEPARTAMENTO         1047249 non-null  object
 1   MUNICIPIO            1047249 non-null  object
 2   CODIGO DANE          1047249 non-null  object
 3   ARMAS MEDIOS         1047249 non-null  object
 4   FECHA HECHO          1047249 non-null  object
 5   GENERO               1047249 non-null  object
 6   GRUPO ETARIO         1046285 non-null  object
 7   DESCRIPCIÓN CONDUCTA  1047249 non-null  object
 8   CANTIDAD             1047249 non-null  int64 
dtypes: int64(1), object(8)
memory usage: 71.9+ MB
```

In [ ]:
```
df.isnull().sum()
```

Out[ ]:
```
DEPARTAMENTO              0
MUNICIPIO                 0
CODIGO DANE               0
ARMAS MEDIOS              0
FECHA HECHO               0
GENERO                    0
GRUPO ETARIO            964
DESCRIPCIÓN CONDUCTA      0
CANTIDAD                  0
dtype: int64
```

In this data set, there are some null values, (not as many as I expected) but the column is a categorical column, so we have to fill these values with some value that we can work with.

Data Cleaning

In [ ]:
```
df['GENERO'].drop_duplicates()
```

Out[ ]:
```
0              FEMENINO
1             MASCULINO
109          NO REPORTA
785327     NO REPORTADO
863052                -
Name: GENERO, dtype: object
```

In [ ]:
```
dict = {'FEMENINO':'femenino',
    'MASCULINO':'masculino',
    'NO REPORTA':'no reporta',
    'NO REPORTADO':'no reporta',
    '-':'no reporta'}
```

In [ ]:
```
df['GENERO'] = df['GENERO'].replace(dict)
```

In [ ]:
```
df['GENERO'].drop_duplicates()
```

Out[ ]:
```
0        femenino
1       masculino
109    no reporta
Name: GENERO, dtype: object
```

In [ ]:
```
df['GRUPO ETARIO'].drop_duplicates()
```

Out[ ]:
```
0               ADULTOS
12         ADOLESCENTES
107             MENORES
132858       NO REPORTA
785327     NO REPORTADO
863052              NaN
Name: GRUPO ETARIO, dtype: object
```

In [ ]:
```
df['GRUPO ETARIO'] = df['GRUPO ETARIO'].fillna('NO REPORTADO')
```

In [ ]:
```
dict_1 = {'ADULTOS':'adultos',
'ADOLESCENTES':'adolescentes',
    'MENORES':'menores',
```

```
    'NO REPORTA': 'no reporta',
    'NO REPORTADO':'no reporta'}
```

In [ ]:
```
rename_dict = {'DEPARTAMENTO':'departamento', 'MUNICIPIO': 'municipio', 'ARMAS MEDIOS'
       'FECHA HECHO':'fecha_hecho', 'DESCRIPCIÓN CONDUCTA':'descripción_conducta',
       'CANTIDAD':'cantidad','GENERO':'genero','GRUPO ETARIO':'grupo_etario'}
```

In [ ]:
```
armas_dict = {'ARMA BLANCA / CORTOPUNZANTE':'cortopunzante',
                            'ARMA DE FUEGO':'arma de fuego',
                            'CONTUNDENTES':'contundentes',
                                    'MOTO':'vehiculo',
                               'NO REPORTA':'no reporta',
              'POLVORA(FUEGOS PIROTECNICOS)':'explosivos',
                                'PUNZANTES':'cortopunzante',
                                 'VEHICULO':'vehiculo',
                              'COMBUSTIBLE':'combustible',
                                  'JERINGA':'material medico',
                                    'PERRO':'animales',
                                'BICICLETA':'vehiculo',
       'ARTEFACTO EXPLOSIVO/CARGA DINAMITA':'explosivos',
                         'MINA ANTIPERSONA':'explosivos',
                       'SUSTANCIAS TOXICAS':'sustacias tóxicas',
                       'SIN EMPLEO DE ARMAS':'sin armas',
                            'AGUA CALIENTE':'casero',
                             'ESCOPOLAMINA':'sustancias tóxicas',
                               'OLLA BOMBA':'explosivos',
                          'GRANADA DE MANO':'explosivos',
                            'PAQUETE BOMBA':'explosivos',
                             'MEDICAMENTOS':'material medico',
                                   'VENENO':'sustancias tóxicas',
                                 'QUIMICOS':'sustancias tóxicas',
                             'CARRO BOMBA':'explosivos',
                                   'GASES':'sustancias tóxicas',
                         'CINTAS/CINTURON':'materiales',
                   'ARTEFACTO INCENDIARIO':'explosivos',
                           'PAPA EXPLOSIVA':'explosivos',
                       'ALIMENTOS VENCIDOS':'sustancias tóxicas',
                        'LICOR ADULTERADO':'sustancias tóxicas',
                                   'ACIDO':'ácido',
                             'ALUCINOGENOS':'sustancias tóxicas',
                               'ALMOHADA':'materiales',
                          'BOLSA PLASTICA':'materiales',
                              'CORTANTES':'cortopunzante',
                               'CUCHILLA':'cortopunzante',
                                'DIRECTA':'materiales',
                           'ARMAS BLANCAS':'cortopunzante',
                       'PRENDAS DE VESTIR':'materiales',
                          'CILINDRO BOMBA':'explosivos',
                                      '-':'no reporta',
                            'NO REPORTADO':'no reporta',
                          'CINTURON BOMBA':'explosivos',
                         'ARMA TRAUMATICA':'contundentes'}
```

In [ ]:
```
df['GRUPO ETARIO'] = df['GRUPO ETARIO'].replace(dict_1)
```

In [ ]:
```
df['GRUPO ETARIO'].drop_duplicates()
```

Out[ ]:
```
0              adultos
12        adolescentes
107            menores
132858      no reporta
Name: GRUPO ETARIO, dtype: object
```

In [ ]:
```
df['FECHA HECHO'] = pd.to_datetime(df['FECHA HECHO'], format='%d/%m/%Y')
```

In [ ]:
```
df.columns
```

Out[ ]:
```
Index(['DEPARTAMENTO', 'MUNICIPIO', 'CODIGO DANE', 'ARMAS MEDIOS',
       'FECHA HECHO', 'GENERO', 'GRUPO ETARIO', 'DESCRIPCIÓN CONDUCTA',
       'CANTIDAD'],
      dtype='object')
```

In [ ]:
```
df = df.rename(columns=(rename_dict))
```

In [ ]:
```
df['armas_medios'].drop_duplicates()
```

Out[ ]:
```
0                   ARMA BLANCA / CORTOPUNZANTE
104                               ARMA DE FUEGO
152                                CONTUNDENTES
358                                        MOTO
359                                  NO REPORTA
364                POLVORA(FUEGOS PIROTECNICOS)
365                                  PUNZANTES
437                                    VEHICULO
554                                 COMBUSTIBLE
630                                     JERINGA
832                                       PERRO
1030                                  BICICLETA
1116      ARTEFACTO EXPLOSIVO/CARGA DINAMITA
1331                          MINA ANTIPERSONA
2034                        SUSTANCIAS TOXICAS
2249                       SIN EMPLEO DE ARMAS
2288                              AGUA CALIENTE
2647                               ESCOPOLAMINA
2648                                OLLA BOMBA
2782                            GRANADA DE MANO
2784                              PAQUETE BOMBA
2994                               MEDICAMENTOS
3726                                     VENENO
4188                                   QUIMICOS
14611                               CARRO BOMBA
26425                                     GASES
31949                          CINTAS/CINTURON
35676                    ARTEFACTO INCENDIARIO
37359                            PAPA EXPLOSIVA
45171                        ALIMENTOS VENCIDOS
68877                          LICOR ADULTERADO
74904                                     ACIDO
76346                              ALUCINOGENOS
187488                                 ALMOHADA
201235                           BOLSA PLASTICA
201309                                CORTANTES
208798                                 CUCHILLA
449553                                  DIRECTA
454954                            ARMAS BLANCAS
601433                         PRENDAS DE VESTIR
706864                           CILINDRO BOMBA
862859                                        -
864668                             NO REPORTADO
942155                           CINTURON BOMBA
960009                           ARMA TRAUMATICA
Name: armas_medios, dtype: object
```

In [ ]:
```
df['armas_medios'] = df['armas_medios'].replace(armas_dict)
```

In [ ]:
```
df['departamento'] = df['departamento'].str.lower()
```

In [ ]:
```
df['municipio'] = df['municipio'].str.lower()
```

In [ ]:
```
df['descripción_conducta'].drop_duplicates()
```

Out[ ]:
```
0                              LESIONES PERSONALES
384    LESIONES CULPOSAS ( EN ACCIDENTE DE TRANSITO )
Name: descripción_conducta, dtype: object
```

In [ ]:
```
desc_dict = {'LESIONES PERSONALES':'lesiones personales', 'LESIONES CULPOSAS ( EN ACCI
```

In [ ]:
```
df['descripción_conducta'] = df['descripción_conducta'].replace(desc_dict)
df['descripción_conducta'].drop_duplicates()
```

Out[ ]:
```
0       lesiones personales
384       lesiones culposas
Name: descripción_conducta, dtype: object
```

In [ ]:
```
df = df.drop(columns=['CODIGO DANE'])
df
```

Out[ ]:

	departamento	municipio	armas_medios	fecha_hecho	genero	grupo_etario	descripción
0	antioquia	girardota	cortopunzante	2010-01-01	femenino	adultos	lesiones
1	antioquia	girardota	cortopunzante	2010-01-01	masculino	adultos	lesiones
2	antioquia	mutatá	cortopunzante	2010-01-01	masculino	adultos	lesiones
3	antioquia	necoclí	cortopunzante	2010-01-01	femenino	adultos	lesiones
4	atlántico	barranquilla (ct)	cortopunzante	2010-01-01	femenino	adultos	lesiones
...	...	...	...	...	...	...	
1047244	cesar	valledupar (ct)	sustancias tóxicas	2022-05-03	masculino	adultos	lesiones
1047245	huila	oporapa	sustancias tóxicas	2022-06-16	femenino	adolescentes	lesiones
1047246	tolima	ibagué (ct)	sustancias tóxicas	2022-04-17	masculino	adultos	lesiones
1047247	cundinamarca	cota	sin armas	2022-03-30	masculino	adultos	lesiones
1047248	cundinamarca	guaduas	sin armas	2022-06-10	masculino	adultos	lesiones

1047249 rows × 8 columns

In [ ]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1047249 entries, 0 to 1047248
Data columns (total 8 columns):
 #   Column                Non-Null Count    Dtype
---  ------                --------------    -----
 0   departamento          1047249 non-null  object
 1   municipio             1047249 non-null  object
 2   armas_medios          1047249 non-null  object
 3   fecha_hecho           1047249 non-null  datetime64[ns]
 4   genero                1047249 non-null  object
 5   grupo_etario          1047249 non-null  object
 6   descripción_conducta  1047249 non-null  object
 7   cantidad              1047249 non-null  int64
dtypes: datetime64[ns](1), int64(1), object(6)
memory usage: 63.9+ MB
```

In [ ]:
```
df.isnull().sum()
```

Out[ ]:
```
departamento          0
municipio             0
armas_medios          0
fecha_hecho           0
genero                0
grupo_etario          0
descripción_conducta  0
cantidad              0
dtype: int64
```

In [ ]:
```
df = df[['fecha_hecho','departamento','municipio','armas_medios','genero','grupo_etari
df
```

Out[ ]:

	fecha_hecho	departamento	municipio	armas_medios	genero	grupo_etario	descripción
0	2010-01-01	antioquia	girardota	cortopunzante	femenino	adultos	lesiones
1	2010-01-01	antioquia	girardota	cortopunzante	masculino	adultos	lesiones
2	2010-01-01	antioquia	mutatá	cortopunzante	masculino	adultos	lesiones
3	2010-01-01	antioquia	necoclí	cortopunzante	femenino	adultos	lesiones
4	2010-01-01	atlántico	barranquilla (ct)	cortopunzante	femenino	adultos	lesiones
...	...	...	...	...	...	...	
1047244	2022-05-03	cesar	valledupar (ct)	sustancias tóxicas	masculino	adultos	lesiones
1047245	2022-06-16	huila	oporapa	sustancias tóxicas	femenino	adolescentes	lesiones
1047246	2022-04-17	tolima	ibagué (ct)	sustancias tóxicas	masculino	adultos	lesiones
1047247	2022-03-30	cundinamarca	cota	sin armas	masculino	adultos	lesiones
1047248	2022-06-10	cundinamarca	guaduas	sin armas	masculino	adultos	lesiones

1047249 rows × 8 columns

At this point, the dataset is more organized and standarized.

EDA

In this project, we are performing exploratory data analysis (EDA) on a dataset in order to extract useful information and insights. To do this, we have defined several functions that allow us to create new columns based on existing data in the dataset.

For example, we may have defined a function to extract the year from a date column, or a function to extract a particular substring from a text column. By creating these new columns, we can gain new insights into the data and answer questions that were previously difficult or impossible to answer.

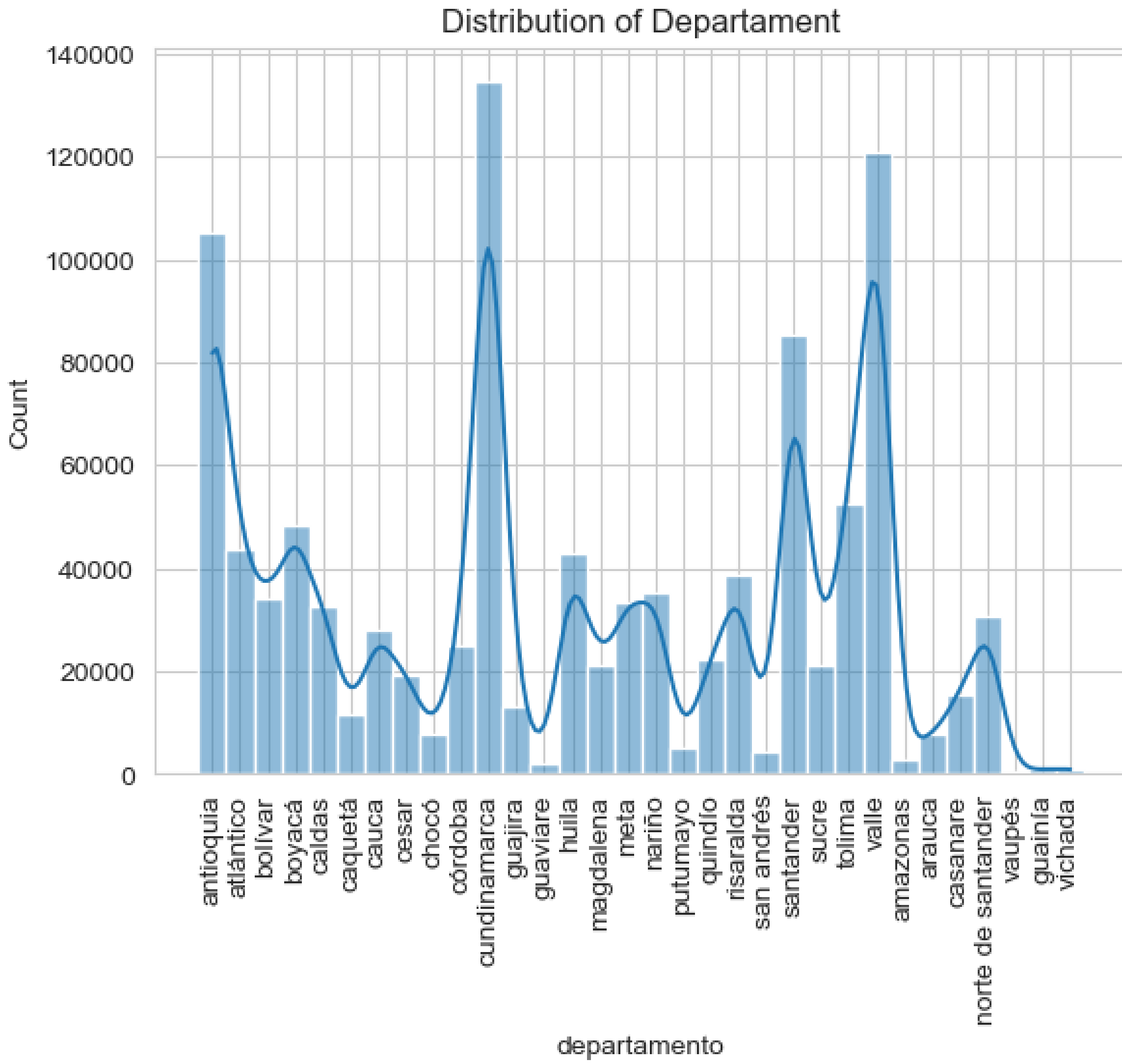
In addition to defining these functions, we are also using various data visualization techniques to explore the data and identify patterns or trends. We may be creating histograms, scatterplots, or other types of plots to help us better understand the relationships between different variables in the dataset.

Overall, the goal of this EDA project is to gain a deeper understanding of the dataset and the underlying processes that generated it. By doing so, we can make more informed decisions and identify opportunities for improvement or optimization.

Anwering the question of accidents by month I define the function 'MES' to obtain the result, subsequently I will plot the result and we can see how is the behavior of the accidents with a line chart.

In [ ]:

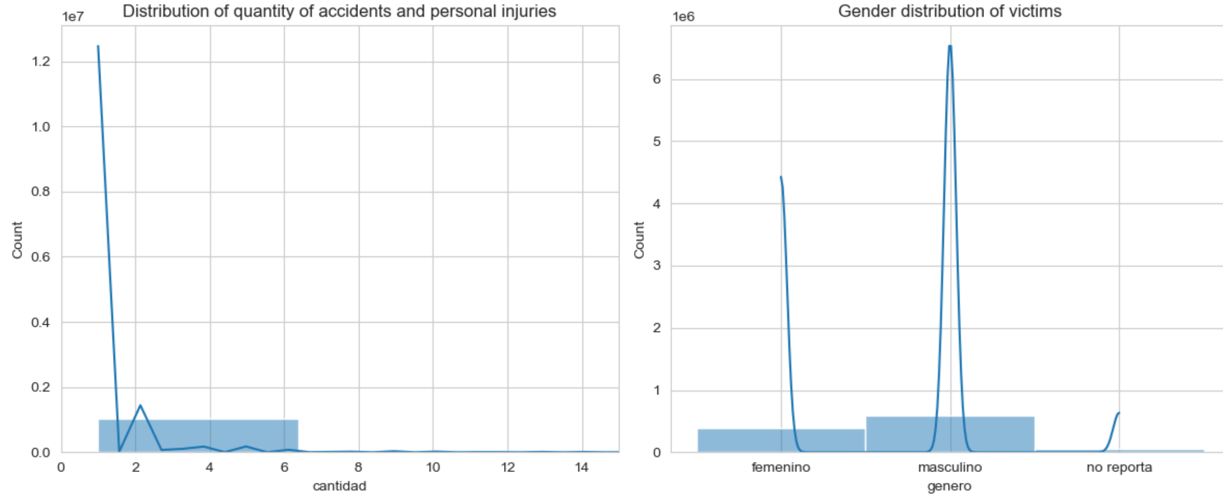
```
# GHistogram
sns.histplot(df, x='departamento', bins=20, kde=True)
plt.title('Distribution of Departament')
plt.xticks(rotation=90)
plt.show()
```



The histogram reveals an interesting distribution of the data, showing a multimodal pattern in the department variable, indicating the presence of several modes in the distribution. The graph also suggests that there are certain departments that occur more frequently than others, leading to the multiple peaks in the histogram. This suggests that there may be underlying factors that contribute to the frequency of accidents in specific departments, which could be further explored in the analysis. Overall, the histogram provides valuable insights into the distribution of accidents across departments and highlights areas that require further investigation.

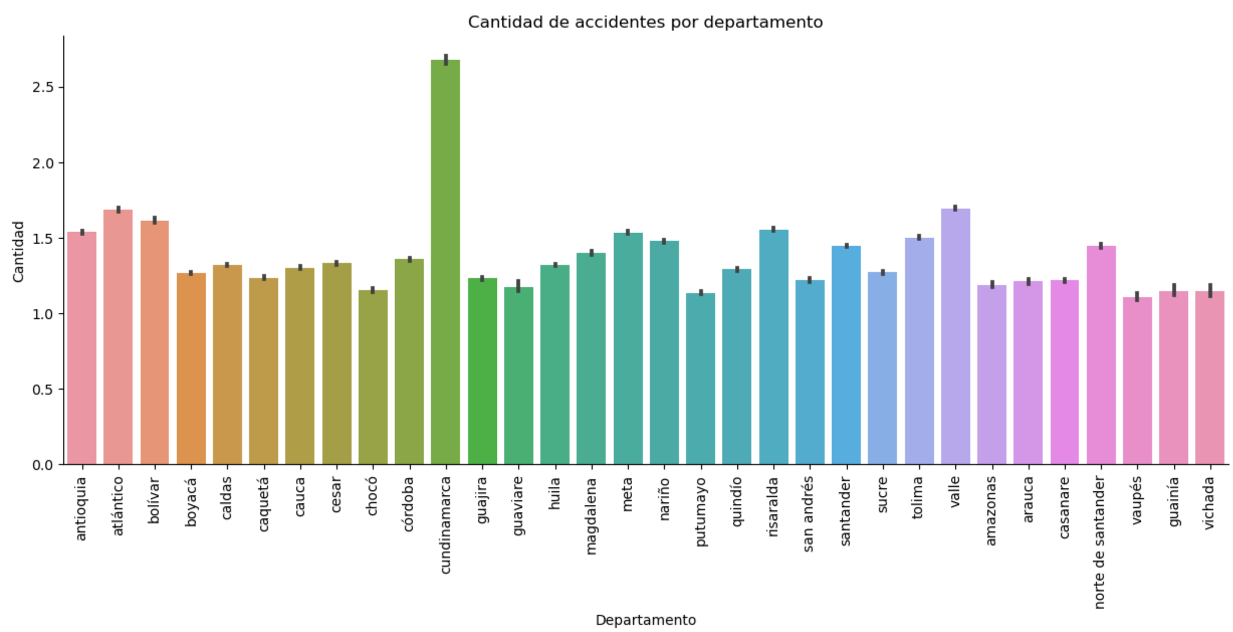
In [ ]:

```
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 5))

sns.histplot(data=df, x="cantidad", ax=ax1, kde=True)
ax1.set_xlim(0, 15)
ax1.set_title("Distribution of quantity of accidents and personal injuries")

sns.histplot(data=df, x="genero", ax=ax2, kde=True)
ax2.set_title("Gender distribution of victims")

plt.tight_layout()
plt.show()
```



These histograms reveal the distribution of the "cantidad" and "genero" variables. We can observe a right-skewed distribution in the "cantidad" histogram, indicating that the most common value is 1, followed by 2. In the "genero" histogram, we can see several peaks that make the graph multimodal. This suggests that there are multiple modes in the gender distribution, which could indicate some underlying patterns or factors affecting the distribution. Further analysis is necessary to fully understand these patterns and their implications.

In [ ]:
```
ax = sns.catplot(data=df, x='departamento', y='cantidad', kind='bar', aspect=2.5)
ax.set(title='Cantidad de accidentes por departamento', xlabel='Departamento', ylabel=
ax.set_xticklabels(rotation=90)
plt.show()
```

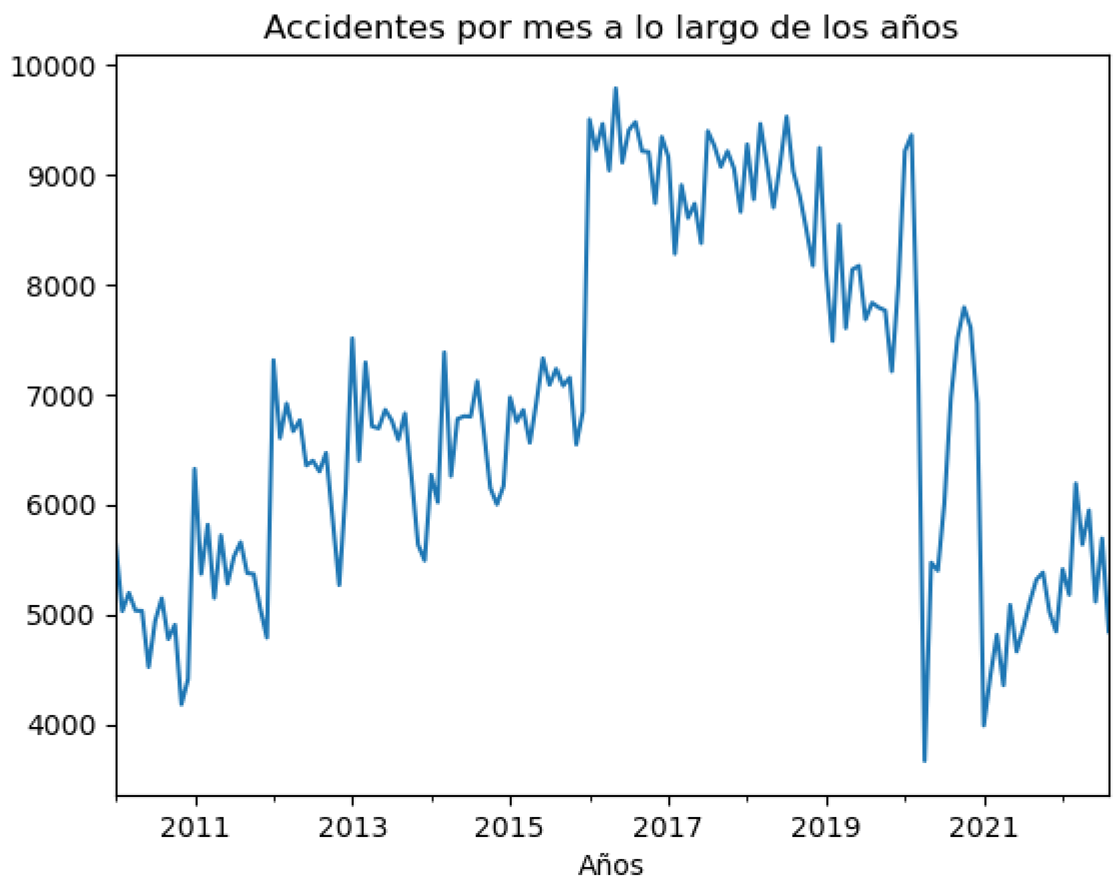


This graph shows the number of accidents by department, where we can see that Cundinamarca is the department with the highest number of accidents, well above the other departments, which are quite similar to each other, and there is not much difference between them. It is interesting to note that this information could be useful for authorities and policymakers to allocate resources and take measures to reduce accidents, especially in Cundinamarca. Further analysis could also be done to explore the possible reasons why Cundinamarca has a higher number of accidents compared to other departments, such as the road infrastructure, population density, and economic activities. Overall, this graph provides valuable insights into the distribution of accidents by department, which could help to improve safety in Colombia.

In [ ]:
```
def MES(df):
    """
    Group accidents by month

    Arguments:
    `df`: A pandas DataFrame

    Outputs:
    `monthly_accidents`: The grouped Series
    """

    # YOUR CODE HERE
    df['fecha_hecho'] = pd.to_datetime(df['fecha_hecho'])
    df['mes'] = df["fecha_hecho"].dt.to_period('M')
    monthly_accidents = df.groupby("mes").size()
    return monthly_accidents
```

In [ ]:
```
MES(df).plot.line()
plt.title('Accidentes por mes a lo largo de los años')
plt.xlabel('Años')
```

Out[ ]:
```
Text(0.5, 0, 'Años')
```



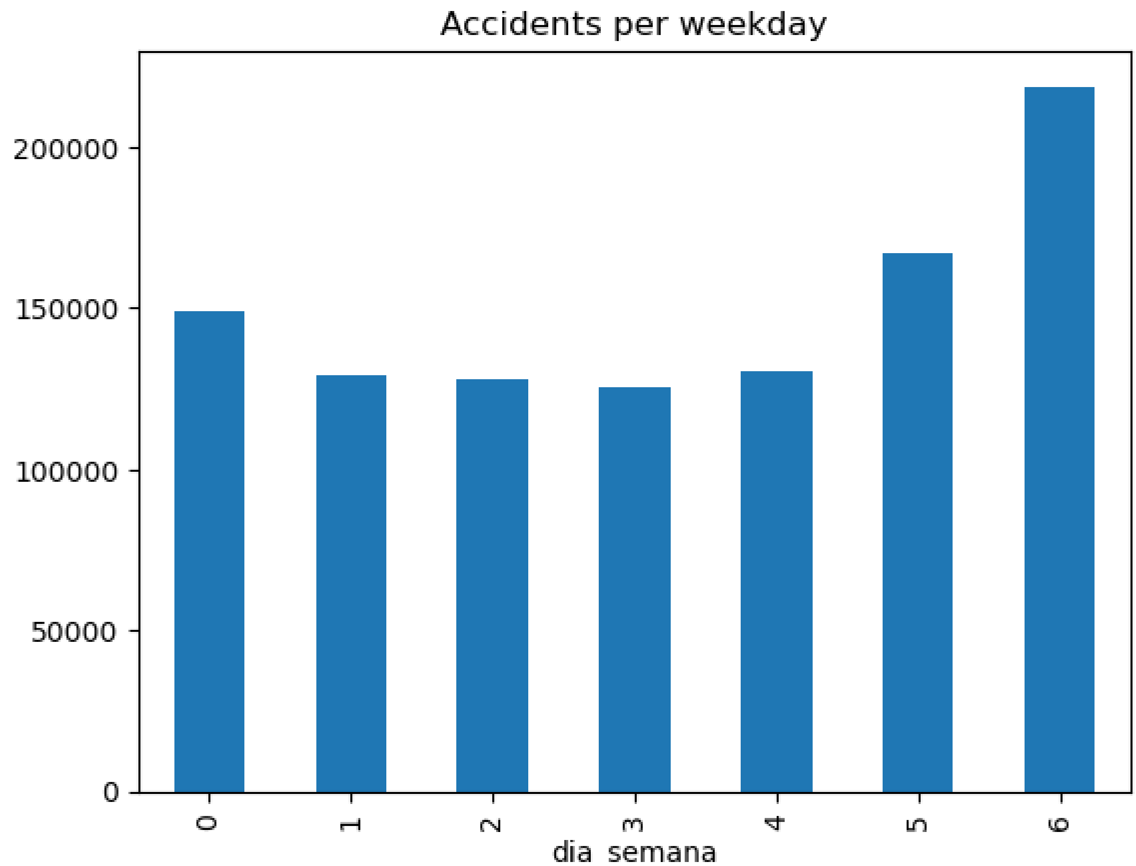
In recent years, accidents have increased significantly, not only in traffic but also in other areas such as the use of bladed weapons or firearms, contact with corrosive acids, among others. This means that more and more people are suffering the tragic effects of accidents, ranging from serious injuries, permanent disability to death. The year 2020 was an exception to this trend, due to the Covid-19 pandemic, which caused a decrease in the number of accidents worldwide. However, early 2021 and 2022 have seen a significant increase in the number of accidents, although they still remain below the numbers of the years prior (to the pandemic). This shows us that there is still a significant risk of suffering an accident, either from traffic, the use of weapons or contact with corrosive acids, so it is important that we all take the necessary measures to prevent these accidents, such as being more aware when driving, comply with speed limits, do not drive under the influence of alcohol or drugs, among other measures. This will help to reduce the number of victims and prevent unnecessary accidents.

Now let´s move on to the behavior for the days of the week.

In [ ]:
```
def DIA(df):
    """
    Group accidents by day of the week

    Arguments:
    `df`: A pandas DataFrame

    Outputs:
    `weekday_accidents`: The grouped Series
    """

    # YOUR CODE HERE
    df['dia_semana'] = pd.to_datetime(df['fecha_hecho']).dt.weekday
    weekday_accidents = df.groupby(["dia_semana"]).size()
    return weekday_accidents
```

In [ ]:
```
DIA(df).plot.bar()
plt.title('Accidents per day of the week')
```

Out[ ]:
```
Text(0.5, 1.0, 'Accidents per weekday')
```

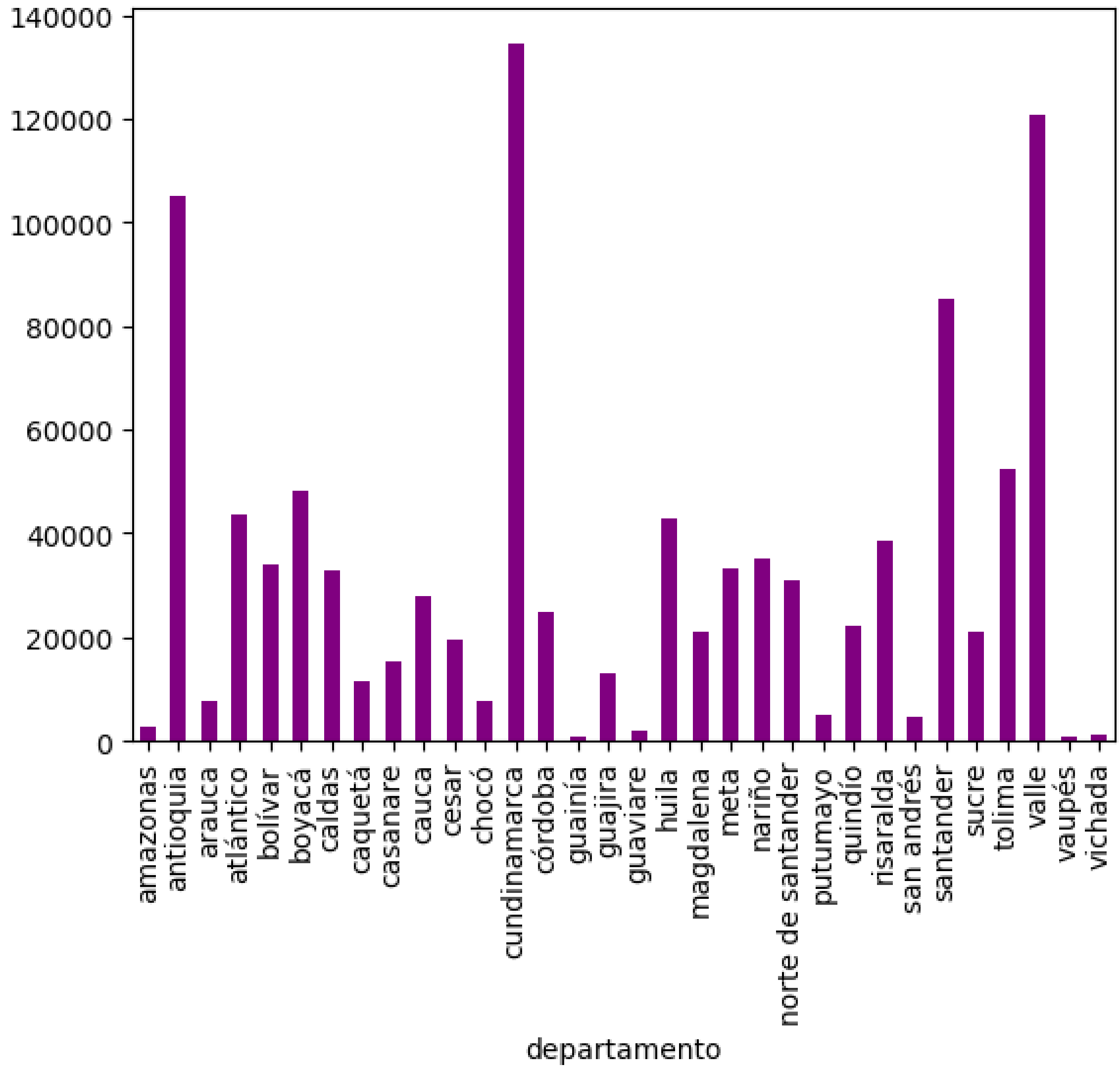


Looking at the bar plot, we can notice a clear pattern: on Mondays, accidents are slightly higher than the rest of the days of the week, while weekends (from Saturday onwards) start to show a significant increase in the number of accidents and personal injuries, with Sunday being the day with the highest accident rate. This leads us to conclude that weekends represent a higher risk of suffering an accident. This could be explained by the excesses to which some people subject themselves during the weekend, such as substance abuse, fights, drunk driving, among others. This is a situation that should be taken into account to prevent these accidents and minimize their impact. Measures should be taken such as the implementation of awareness campaigns on the risks of drunk driving, substance abuse and the use of weapons. Stricter controls should also be implemented to prevent excessive use of substances, use of weapons, and speeding. These measures will help prevent these accidents and save lives.

In [ ]:
```
def DEPARTAMENTO(df):
    """
    Group accidents by borough

    Arguments:
    `df`: A pandas DataFrame

    Outputs:
    `boroughs`: The grouped Series
    """

    # YOUR CODE HERE
    df['departamento'].drop_duplicates()
    boroughs = df.groupby(['departamento']).size()
    return boroughs
```

In [ ]:
```
DEPARTAMENTO(df).plot.bar(color='purple')
```

Out[ ]:
```
<AxesSubplot:xlabel='departamento'>
```



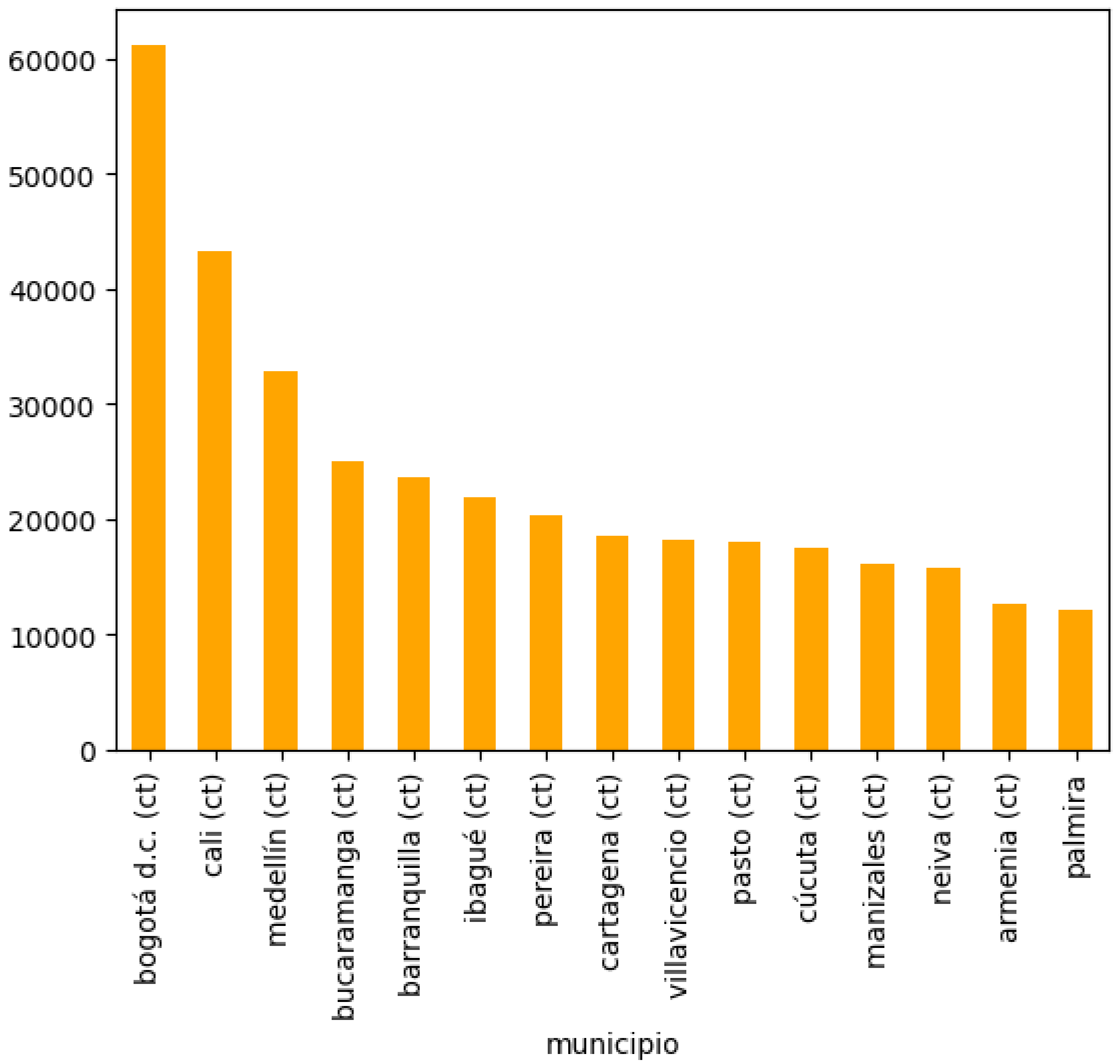
Looking at the graph above, we can see that the Colombian departments with the highest accident rate are Cundinamarca, Valle, Antioquia, Santander and Tolima. This leads us to conclude that these five departments, despite representing only 20% of the Colombian population, account for almost 40% of the accidents, which shows that the risk of suffering an accident is higher in these departments. This can be explained by the lack of adequate infrastructure for transportation, lack of awareness of drivers, speeding, substance abuse, handling and carrying weapons, lack of citizen awareness, among other factors. These are risks that must be taken into account to prevent these accidents and save lives. In addition, it is necessary for local governments to implement awareness campaigns on the risks of drunk driving, substance abuse, and the carrying of firearms and weapons, to promote a culture of tolerance, as well as stricter controls to minimize the number of accidents.

In [ ]:
```
def MUNICIPIO(df):
    """
    Group accidents by borough
```

```
    Arguments:
    `df`: A pandas DataFrame

    Outputs:
    `boroughs`: The grouped Series
    """

    # YOUR CODE HERE
    df['municipio'].drop_duplicates()
    boroughs = df.groupby(['municipio']).size()
    boroughs = boroughs.sort_values(ascending=False).head(15)
    return boroughs
```

In [ ]:
```
MUNICIPIO(df).plot.bar(color='orange')
```

Out[ ]: <AxesSubplot:xlabel='municipio'>

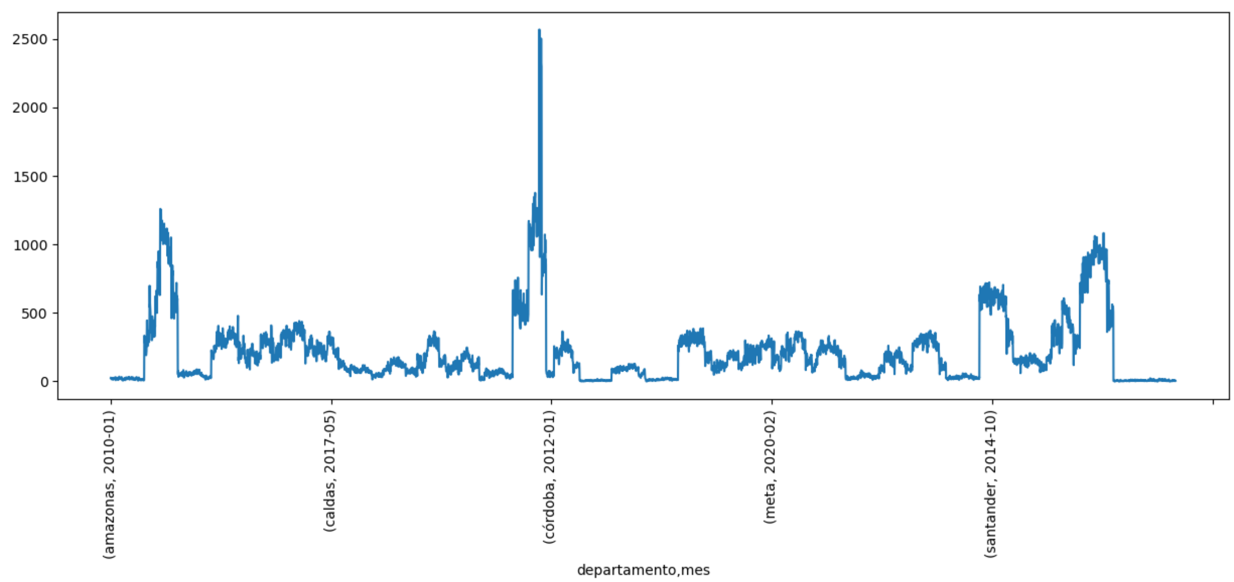


The bar chart above shows the municipalities in Colombia with the highest number of accidents and personal injuries. The chart indicates that the top five municipalities with the highest number of accidents and personal injuries are Bogotá D.C., Cali, Medellín, Bucaramanga, and Barranquilla.

the bar chart provides a high-level overview of the municipalities with the highest number of accidents and personal injuries in Colombia. This information can be used to develop targeted interventions to reduce the number of accidents and injuries in these areas, ultimately leading to a safer and more secure society for all.

In [ ]:
```
def MES_DEPAR(df):
    """
    Calculate accidents per hour for each borough

    Arguments:
    `df`: A pandas DataFrame


    Outputs:
    `bor_hour`: A Series. This should be the result of doing groupby by borough
    and hour.
    """

    # YOUR CODE HERE
    df['mes'] = pd.to_datetime(df['fecha_hecho']).dt.to_period('M')
    bor_hour= df.groupby(["departamento","mes"]).size()

    return bor_hour
```

In [ ]:
```
fig, ax = plt.subplots(figsize=(15,5))
ax = MES_DEPAR(df).plot()
plt.xticks(rotation=90)
plt.show()
```



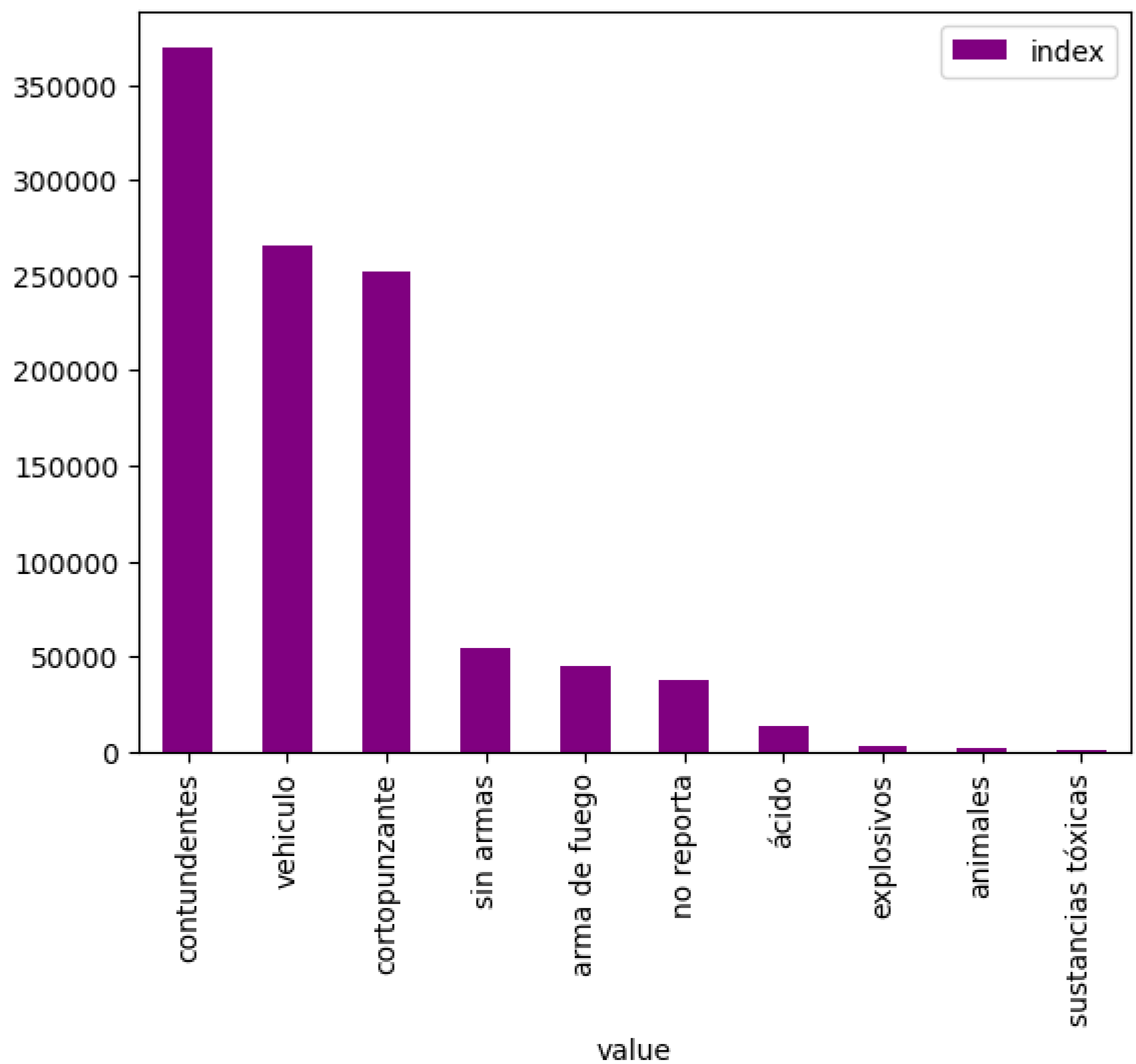
The line chart above shows the trends of accidents and personal injuries over time, specifically focusing on the year with the highest number of cases in the Department of Córdoba. The chart indicates that the year with the highest number of accidents and personal injuries in the Department of Córdoba was 2012.

In [ ]:
```
def FACTORES(df):
    """
    Finds which 6 factors cause the most accidents, without
```

```
    double counting the contributing factors of a single accident.

    Arguments:
    `contrib_df`: A pandas DataFrame.

    Outputs:
    `factors_most_acc`: A pandas DataFrame. It has only 10 elements, which are,
    sorted in descending order, the contributing factors with the most accidents.
    The column with the actual numbers is named `index`.
    """

    # YOUR CODE HERE

    contrib_df = pd.melt(df.reset_index(),id_vars ="index", value_vars= 'armas_medios'
    contrib_df= contrib_df.drop(columns=['variable'])
    contrib_df = contrib_df.drop_duplicates(keep='first')
    factors_most_acc = contrib_df.groupby('value').count().sort_values(by='index', asc
    factors_most_acc= factors_most_acc.head(10)

    return factors_most_acc
```

In [ ]:
```
FACTORES(df).plot.bar(color='purple')
```

Out[ ]: <AxesSubplot:xlabel='value'>

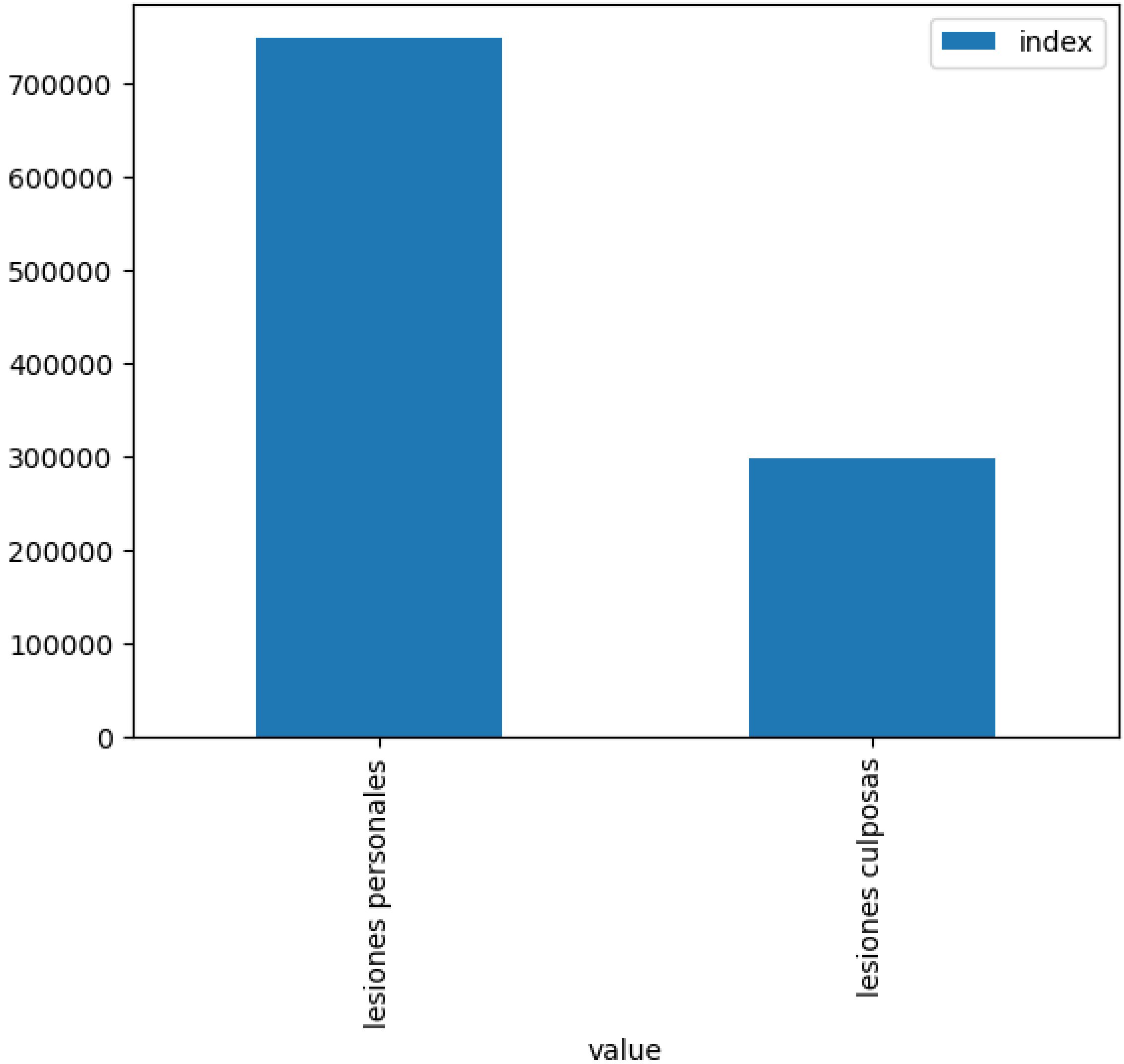


The bar chart above displays the 10 most common factors associated with accidents in Colombia. According to the chart, the most frequent factor is blunt objects, which suggests that most frequent personal injurie in Colombia result from one person striking another with a non-cutting weapon. The second most common factor is vehicular accidents, which is not surprising given the high number of cars and motorcycles in the country. The third most frequent factor is sharp-edged weapons such as knives and machetes.

In [ ]:
```
def CONDUCTA(df):
    """
    Finds which 6 factors cause the most accidents, without
    double counting the contributing factors of a single accident.

    Arguments:
    `contrib_df`: A pandas DataFrame.

    Outputs:
    `contrib_df`: A pandas DataFrame. It has only 6 elements, which are,
    sorted in descending order, the contributing factors with the most accidents.
    The column with the actual numbers is named `index`.
    """

    # YOUR CODE HERE

    contrib_df = pd.melt(df.reset_index(),id_vars ="index", value_vars= 'descripción_c
    contrib_df= contrib_df.drop(columns=['variable'])
    contrib_df = contrib_df.drop_duplicates(keep='first')
    conduct_acc = contrib_df.groupby('value').count().sort_values(by='index', ascendir

    return conduct_acc
```

In [ ]:
```
CONDUCTA(df).plot(kind='bar')
```

Out[ ]:
```
<AxesSubplot:xlabel='value'>
```



Based on the information provided, it appears that the majority of the cases are accidents caused by either the injured person's carelessness or unintentional harm caused by another person. However, there is a significant percentage of cases that are caused intentionally by other individuals with the intention to harm or even take the victim's life. Additionally, some cases may reflect attempted suicides, although it is unclear from the given information.

It is important to note that speculation without sufficient evidence can be misleading and potentially harmful. Therefore, it is necessary to further investigate and analyze the data to accurately determine the causes of these accidents and take appropriate measures to prevent them.

In [ ]:
```
contingency = pd.crosstab(columns=df['armas_medios'],index=df['departamento'])
contingency = contingency
contingency
```

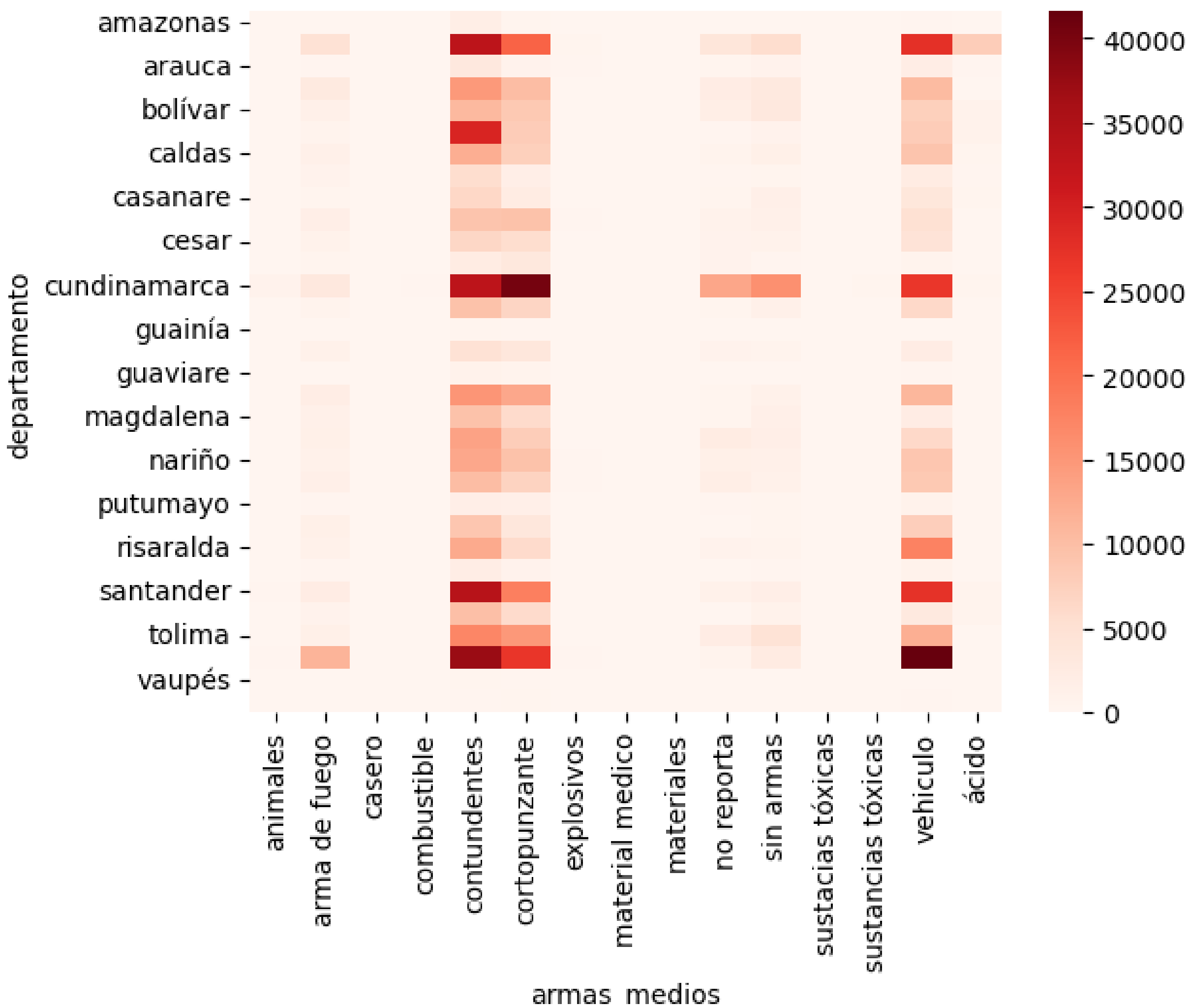
Out[ ]:

armas_medios	animales	arma de fuego	casero	combustible	contundentes	cortopunzante	explosivos	mate med
departamento								
amazonas	18	57	2	1	1937	453	3	
antioquia	154	4725	33	72	33281	21509	443	
arauca	7	260	2	7	3104	701	175	
atlántico	30	2849	45	20	14714	10131	35	
bolívar	2	1165	23	54	10574	8606	31	
boyacá	86	522	18	17	29376	8016	23	
caldas	69	1142	9	14	12127	7599	33	
caquetá	17	737	7	1	5611	1774	133	
casanare	17	376	5	9	6466	2343	19	
cauca	41	1643	20	21	9187	9433	318	
cesar	3	871	4	11	6571	5555	42	
chocó	3	462	4	4	2416	3284	50	
cundinamarca	698	3224	74	183	33154	40321	140	
córdoba	9	612	9	12	9547	6682	63	
guainía	1	2	0	1	454	291	2	
guajira	0	1094	6	5	4866	3554	42	
guaviare	0	89	0	1	964	619	35	
huila	120	1993	17	25	15120	12873	152	
magdalena	10	1298	21	18	9422	5972	30	
meta	60	1341	11	33	13514	7925	131	
nariño	78	997	6	17	12928	9492	282	
norte de santander	31	1496	9	23	10168	7142	313	
putumayo	5	280	3	2	1741	1511	94	
quindío	71	1371	9	11	8791	3508	13	
risaralda	56	1120	17	10	12533	5969	18	
san andrés	3	264	1	3	2039	721	0	
santander	198	2117	44	38	33861	18091	61	
sucre	7	743	10	11	9872	5937	14	
tolima	55	1373	17	35	17106	14812	60	
valle	178	11324	37	54	37226	26852	245	
vaupés	0	7	0	0	342	229	0	

armas_medios	animales	arma de fuego	casero	combustible	contundentes	cortopunzante	explosivos	mate med
departamento								
vichada	1	28	1	2	311	462	4	

In [ ]:
```python
sns.heatmap(contingency, cmap='Reds')
```

Out[ ]: <AxesSubplot:xlabel='armas_medios', ylabel='departamento'>



The previous heat map shows that regardless of the department in Colombia, the main causes of accidents and personal injuries are blunt and sharp-edged weapons, as well as vehicular accidents. This information can be used to develop policies and measures to prevent such accidents. However, a more in-depth study is required to determine whether the causes are intentional, such as crimes or attempted homicides, or unintentional accidents without any intent to harm.

Although the data does not reveal the cause of the accidents, we can infer that street fights and robbery attempts are the most common reasons for such incidents. It is important to note that speculations without sufficient evidence can be misleading and potentially harmful. Therefore, further investigation and analysis of the data are necessary to determine the root cause of these accidents accurately.

Based on the heat map, it is evident that there is a need for more focused efforts to prevent the use of blunt and sharp-edged weapons in crimes and reduce the number of vehicular accidents. This information can be used to develop targeted policies to improve public safety and reduce the number of accidents and injuries in the country.

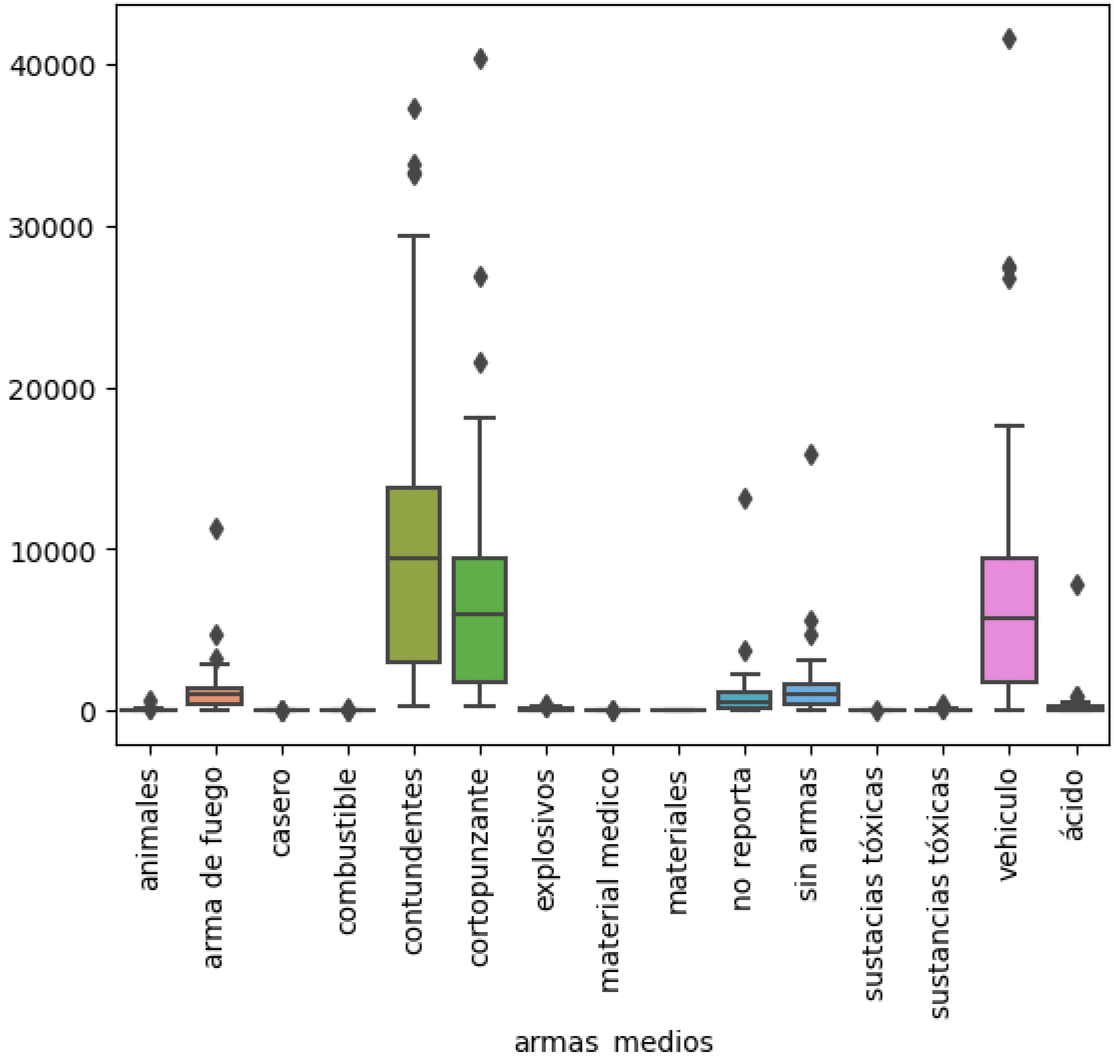
In [ ]:
```
con_df = pd.melt(contingency.reset_index(), id_vars =['departamento'], value_vars =['a
                'arma de fuego' ,'casero',      'combustible',
                'contundentes', 'cortopunzante',        'explosivos',   'material medi
                    'sin armas',        'sustacias tóxicas',    'sustancias tóxicas',
                'vehiculo',     'ácido'],
              var_name ='medios', value_name ='values')
con_df
```

Out[ ]:

	departamento	medios	values
0	amazonas	animales	18
1	antioquia	animales	154
2	arauca	animales	7
3	atlántico	animales	30
4	bolívar	animales	2
...	...	...	...
475	sucre	ácido	515
476	tolima	ácido	73
477	valle	ácido	64
478	vaupés	ácido	1
479	vichada	ácido	7

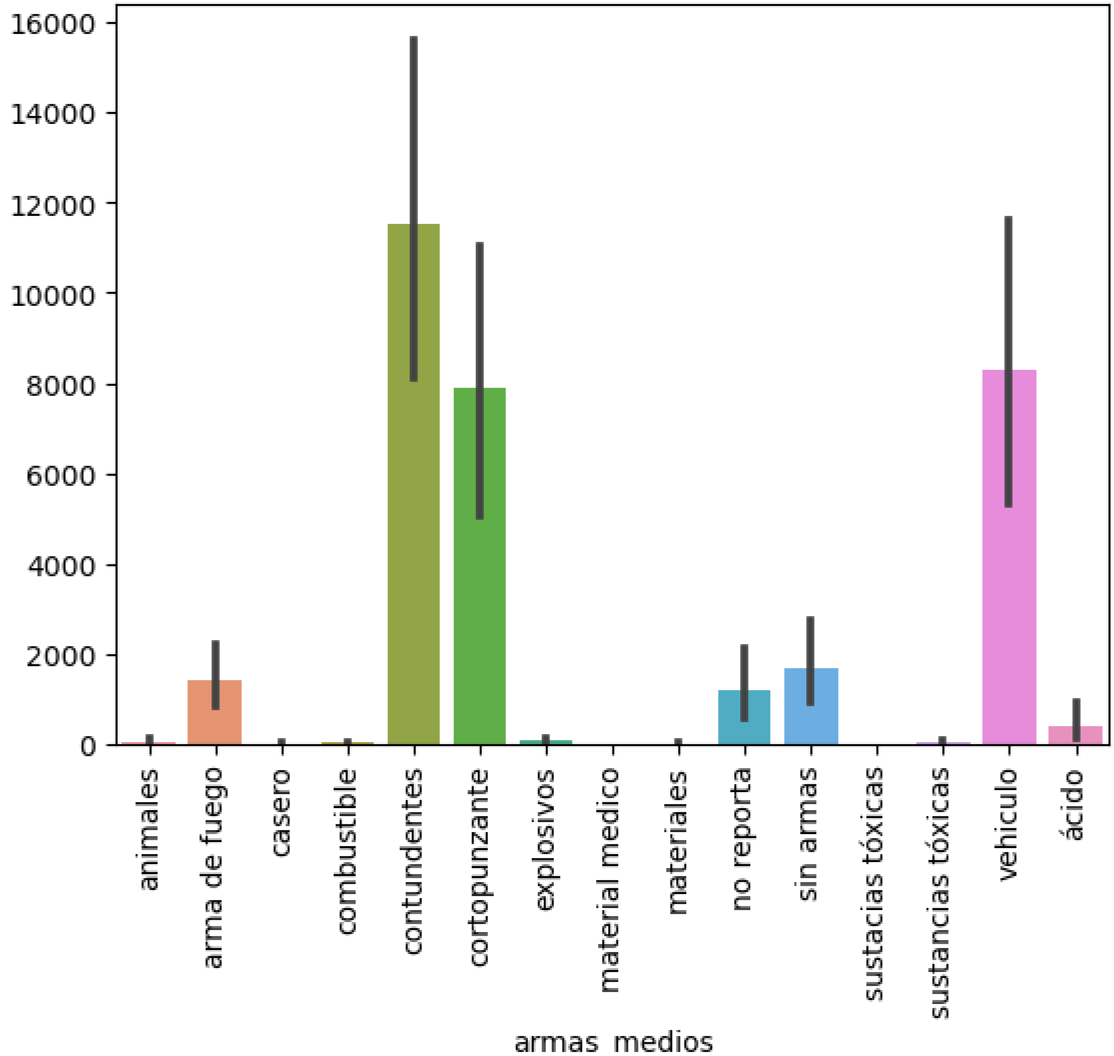
480 rows × 3 columns

In [ ]:
```
sns.boxplot(contingency)
plt.xticks(rotation=90)
plt.show()
```



In [ ]:

```
sns.barplot(data=contingency)
plt.xticks(rotation=90)
plt.show()
```



The previous charts confirm what the heatmap revealed earlier. Here we can observe the distribution of the most frequent causes of accidents and personal injuries throughout the entire country. As previously mentioned, blunt and sharp-edged weapons, as well as vehicular accidents, are the most common causes in most departments.

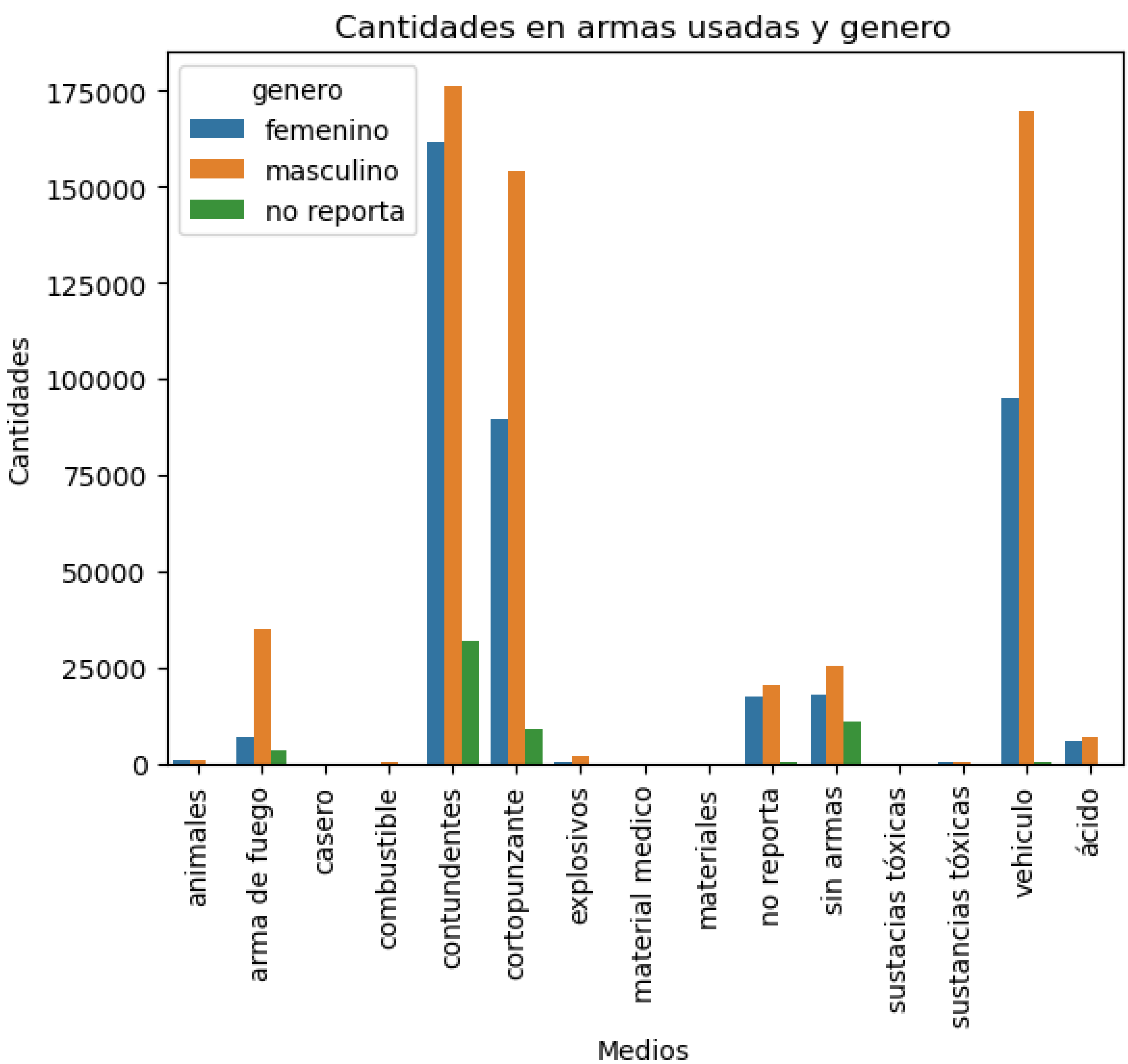
In [ ]:
```
def armas_gen(df):
    """
    df: pandas dataframe

    arguments:
    output: pandas dataframe, it has only 3 columns
    gender, means to make accident and quantity

    """
    contingency_2 = pd.crosstab(index=df['genero'],columns=df['armas_medios'])
    d_f = pd.melt(contingency_2.reset_index(), id_vars =['genero'], value_vars =['anir
                'arma de fuego' ,'casero',      'combustible',
                'contundentes', 'cortopunzante',        'explosivos',   'material medi
                    'sin armas',        'sustacias tóxicas',    'sustancias tóxicas',
                'vehiculo',     'ácido'],
              var_name ='medios', value_name ='cantidades')
    return d_f
```

In [ ]:
```
armas_gen(df).head()
```

Out[ ]:

	genero	medios	cantidades
0	femenino	animales	904
1	masculino	animales	908
2	no reporta	animales	216
3	femenino	arma de fuego	6927
4	masculino	arma de fuego	34929

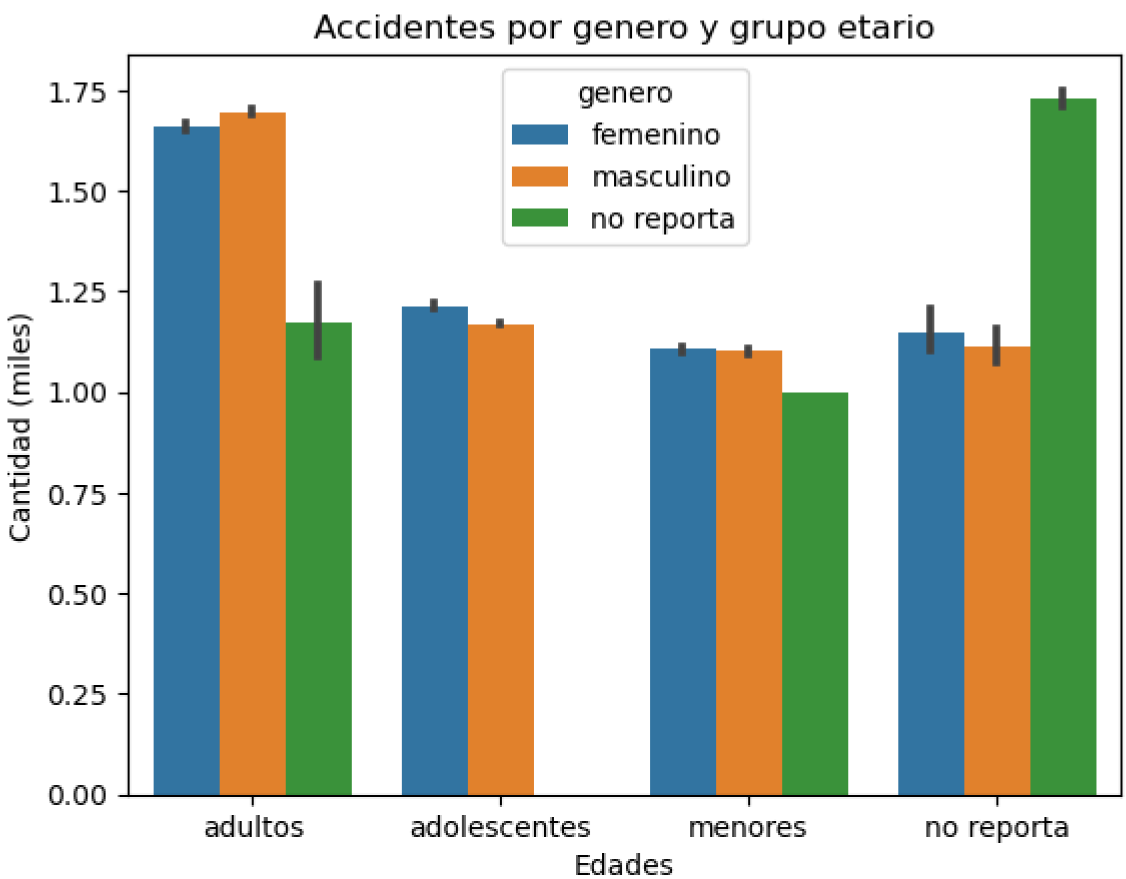
In [ ]:
```
sns.barplot(data=armas_gen(df), x='medios',y='cantidades',hue='genero')
plt.xticks(rotation=90)
plt.title('Cantidades en armas usadas y genero')
plt.xlabel('Medios')
plt.ylabel('Cantidades')
plt.show()
```



The previous graph shows the relationship between the gender of the victim and the type of weapon used in causing the injury. It is evident that adult males are the most affected by both vehicular accidents and personal injuries caused by sharp-edged and blunt objects. However, it is interesting to note that blunt objects affect women to a much greater extent than sharp-edged weapons, suggesting a high level of female involvement in violence. This could be attributed to domestic violence, which is a common cause of injury among women. Further analysis is needed to determine the root causes of violence against women in Colombia and to develop effective policies to prevent it.

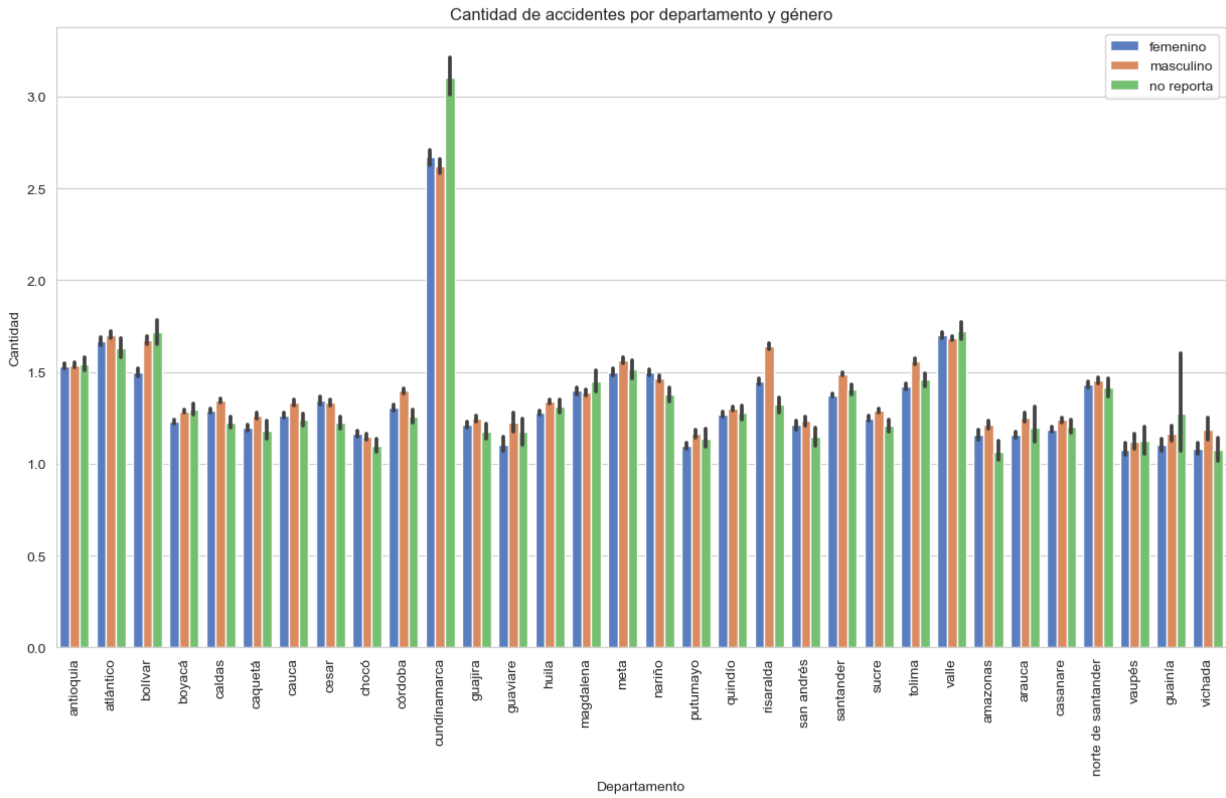
In [ ]:
```
sns.barplot(data=df,x='grupo_etario',y='cantidad',hue='genero')
plt.title('Accidentes por genero y grupo etario')
plt.xlabel('Edades')
plt.ylabel('Cantidad (miles)')
```

Out[ ]:
```
Text(0, 0.5, 'Cantidad (miles)')
```



Looking at the bar graph, it can be seen that accidents affect men and women in a very similar way, with adults being the most affected group. Although data on the gender and age of those affected are not reported, we know that these accidents also affect children and adolescents, which represents a large number of people who have suffered some kind of accident. This leads us to conclude that the factors contributing to the number of accidents are not limited to a single gender or age, but are many and varied. Among them, we can highlight inappropriate driving behavior, such as driving under the influence of alcohol, speeding, substance abuse, use of weapons, among others. These factors, together with the lack of awareness and respect for traffic rules, contribute to the increase in accidents shown in the graph. Therefore, it is necessary that we all become aware of the dangers and risks involved in not respecting the rules, in order to prevent these accidents and save lives.

In [ ]:
```
sns.set_style("whitegrid")
plt.figure(figsize=(15, 8))
ax = sns.barplot(data=df, x="departamento", y='cantidad',hue="genero", palette="muted'
ax.set_title('Cantidad de accidentes por departamento y género')
ax.set_xlabel('Departamento')
ax.set_ylabel('Cantidad')
ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
plt.legend(loc='upper right')
plt.show()
```



Analyzing the bar chart, we can clearly observe that the department of Cundinamarca has the highest number of accidents and personal injuries in Colombia. Although women are more affected, the difference between genders is not statistically significant. However, it is concerning that there are a significant number of accidents where the gender of the victim was not reported. This makes it difficult to determine the true gender distribution of the victims, which is crucial information for designing effective interventions.

We can also see that the number of accidents and personal injuries is quite high, which is alarming. This could be attributed to various factors such as the high rate of violence in Bogotá city or poor road safety measures. It is important to consider the underlying causes of these accidents to develop more effective prevention strategies.

Overall, the data provides valuable insights into the current situation in Colombia regarding accidents and personal injuries. However, further analysis is required to gain a deeper understanding of the issue and to design effective interventions to reduce the number of accidents and injuries.

In [ ]:
```
df = df.set_index('fecha_hecho').reset_index()
df
```

Out[ ]:

	fecha_hecho	departamento	municipio	armas_medios	genero	grupo_etario	descripción
0	2010-01-01	antioquia	girardota	cortopunzante	femenino	adultos	lesiones
1	2010-01-01	antioquia	girardota	cortopunzante	masculino	adultos	lesiones
2	2010-01-01	antioquia	mutatá	cortopunzante	masculino	adultos	lesiones
3	2010-01-01	antioquia	necoclí	cortopunzante	femenino	adultos	lesiones
4	2010-01-01	atlántico	barranquilla (ct)	cortopunzante	femenino	adultos	lesiones
...	...	...	...	...	...	...	
1047244	2022-05-03	cesar	valledupar (ct)	sustancias tóxicas	masculino	adultos	lesiones
1047245	2022-06-16	huila	oporapa	sustancias tóxicas	femenino	adolescentes	lesiones
1047246	2022-04-17	tolima	ibagué (ct)	sustancias tóxicas	masculino	adultos	lesiones
1047247	2022-03-30	cundinamarca	cota	sin armas	masculino	adultos	lesiones
1047248	2022-06-10	cundinamarca	guaduas	sin armas	masculino	adultos	lesiones

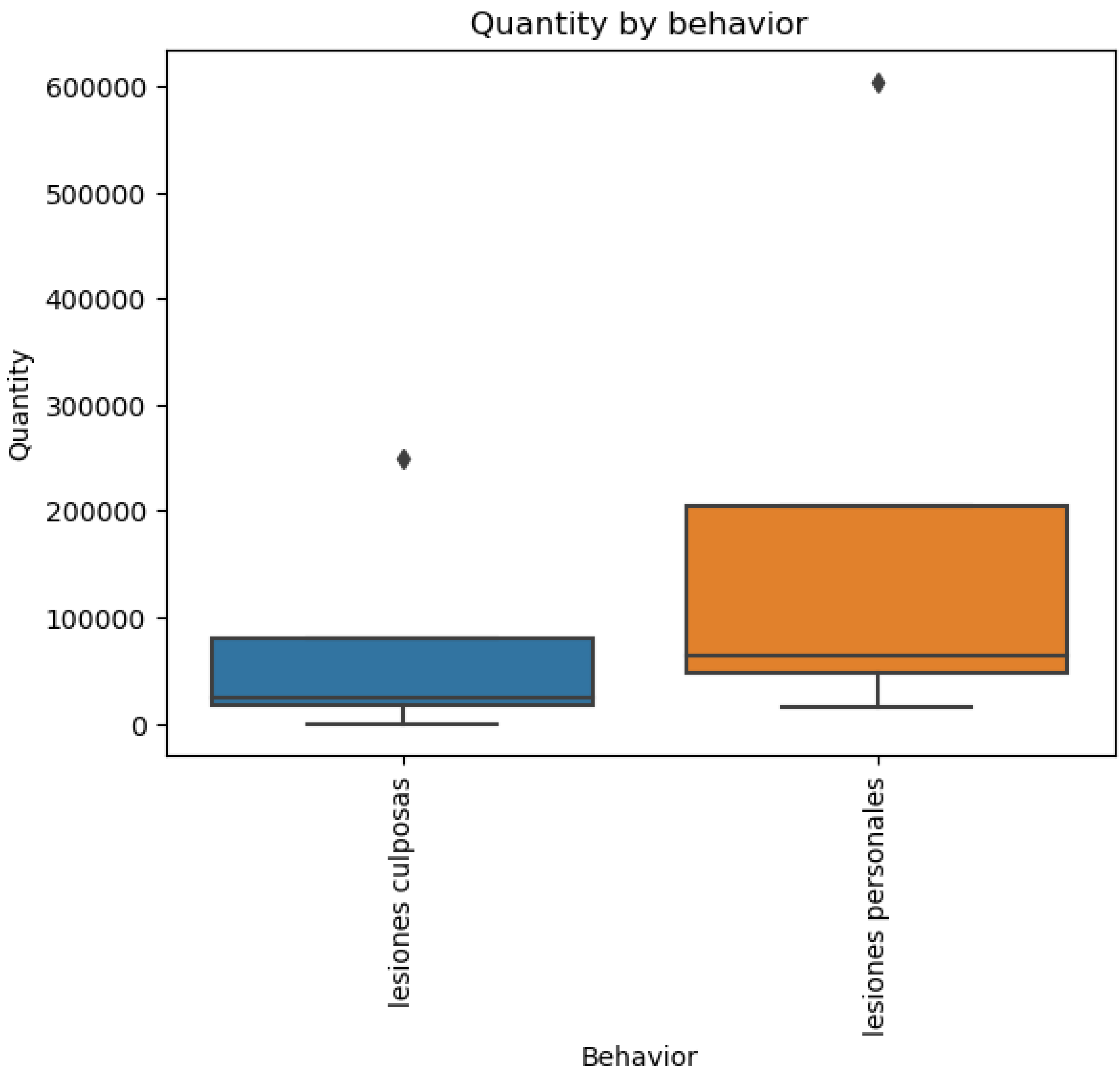
1047249 rows × 10 columns

In [ ]:
```
con = pd.crosstab(index=df['grupo_etario'],columns= df['descripción_conducta'])
con
```

Out[ ]:

descripción_conducta	lesiones culposas	lesiones personales
grupo_etario		
adolescentes	25284	70479
adultos	249258	604306
menores	24112	15835
no reporta	55	57920

In [ ]:
```
sns.boxplot(data=con)
plt.xticks(rotation=90)
plt.title('Quantity by behavior')
plt.xlabel('Behavior')
plt.ylabel('Quantity')
plt.show()
```



Personal injuries are a significant part of accidental injuries, which suggests that it is more common for injuries to be reported as having no apparent intention to cause harm than the opposite. This may indicate that a large number of injuries are due to negligence or carelessness rather than intentional harm. It also highlights the importance of prevention strategies and safety measures to reduce the number of accidents and personal injuries that occur. Furthermore, accurate reporting of the causes of injuries is essential to develop effective policies and programs to prevent and reduce the incidence of personal injuries.

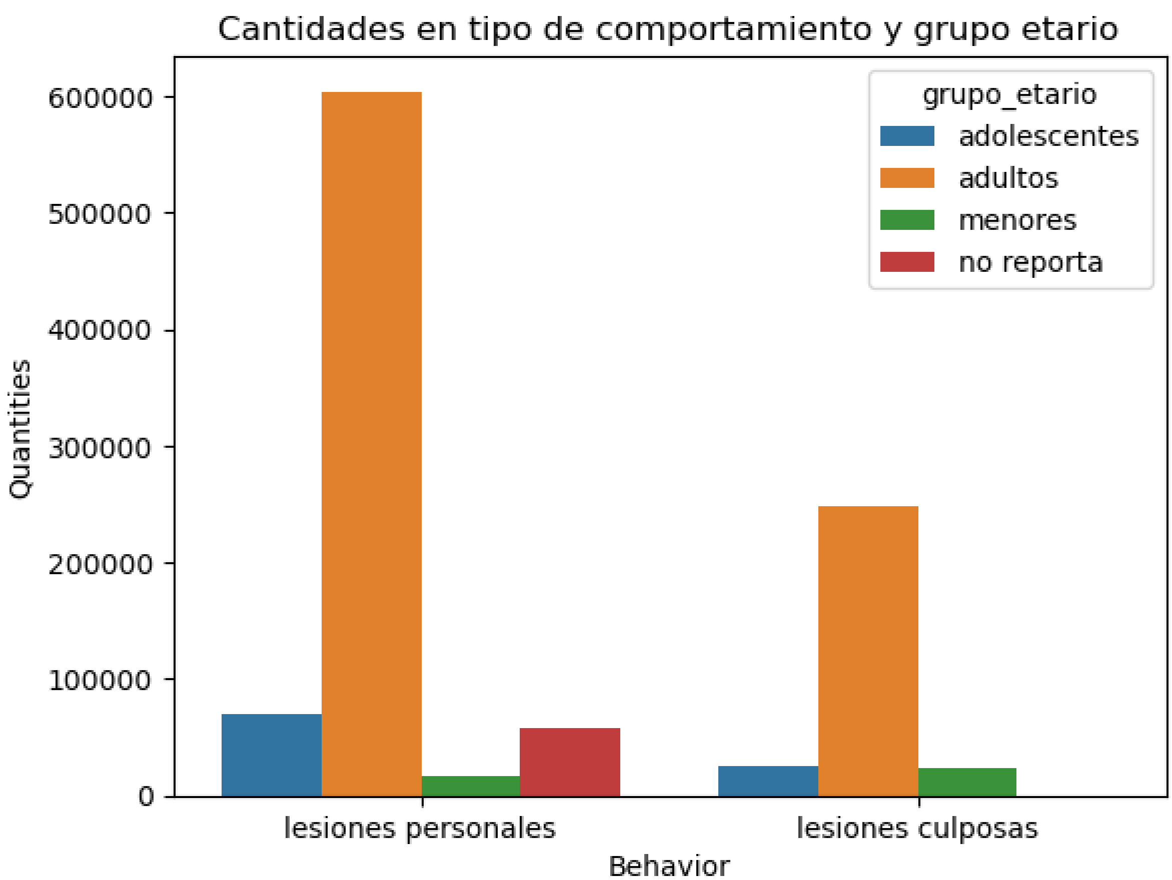
In [ ]:
```
con_melted = pd.melt(con.reset_index(), id_vars =['grupo_etario'], value_vars =['lesio
              var_name ='causas', value_name ='value')

con_melted
```

Out[ ]:

	grupo_etario	causas	value
0	adolescentes	lesiones personales	70479
1	adultos	lesiones personales	604306
2	menores	lesiones personales	15835
3	no reporta	lesiones personales	57920
4	adolescentes	lesiones culposas	25284
5	adultos	lesiones culposas	249258
6	menores	lesiones culposas	24112
7	no reporta	lesiones culposas	55

In [ ]:

```
sns.barplot(data=con_melted, x='causas',y='value',hue='grupo_etario')
plt.title('Cantidades en tipo de comportamiento y grupo etario')
plt.xlabel('Behavior')
plt.ylabel('Quantities')
plt.show()
```



The personal injuries data shows a significant gender gap, with men being affected more than women. This could be due to the fact that men are more exposed to heavy or high-risk jobs, or they are simply more likely to take risks than women. In terms of intentional or unintentional injuries caused by others, it could also be due to their greater involvement in street violence.

It's worth noting that these gender differences in personal injuries are not absolute and can vary depending on the type of injury and the context in which it occurs. However, this data does suggest that there may be certain gender-specific factors that contribute to personal injury rates. This underscores the need for gender-sensitive policies and interventions aimed at preventing and addressing personal injuries, particularly among men.

In [ ]:
```
df.to_csv("C:/Users/Jorge/Downloads/Projects/colombian_acc.csv",encoding = 'utf-8')
```

KMEANS

For this dataset We have decided to use KMeans algorithm to cluster the data and understand the performance of each group.

In [ ]:
```
# Import libraries for create the model
import base64
from pylab import rcParams # For the size of plots
from sklearn import preprocessing # Library to transfor the data

# Libraries for the model
from sklearn.cluster import KMeans
from sklearn.metrics import f1_score
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
```

We drop the columns we don't need.

In [ ]:
```
df1 = df.copy()
df1 = df1.drop(columns=['fecha_hecho','mmunicipio','mes','dia_semana']) # In order to
df1
```

Out[ ]:

	departamento	armas_medios	genero	grupo_etario	descripción_conducta	cantidad
0	antioquia	cortopunzante	femenino	adultos	lesiones personales	2
1	antioquia	cortopunzante	masculino	adultos	lesiones personales	1
2	antioquia	cortopunzante	masculino	adultos	lesiones personales	1
3	antioquia	cortopunzante	femenino	adultos	lesiones personales	1
4	atlántico	cortopunzante	femenino	adultos	lesiones personales	2
...	...	...	...	...	...	...
1047244	cesar	sustancias tóxicas	masculino	adultos	lesiones personales	1
1047245	huila	sustancias tóxicas	femenino	adolescentes	lesiones personales	1
1047246	tolima	sustancias tóxicas	masculino	adultos	lesiones personales	1
1047247	cundinamarca	sin armas	masculino	adultos	lesiones personales	1
1047248	cundinamarca	sin armas	masculino	adultos	lesiones personales	1

1047249 rows × 6 columns

Transform the categorical values into numerical values.

In [ ]:

```
CATEGORICAL_COLUMNS = ['departamento','armas_medios','genero','grupo_etario','descripc
# Iterate with each object type column and transform it in categorical type to obtaine
for column in CATEGORICAL_COLUMNS:
  df1[column] = df1[column].astype('category').cat.codes
  df1[column] = df1[column].astype('float64')
```

We get the Float values for each column, Now we can normalize the data in order to feed the model.

In [ ]:

```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1047249 entries, 0 to 1047248
Data columns (total 6 columns):
 #   Column                Non-Null Count    Dtype
---  ------                --------------    -----
 0   departamento          1047249 non-null  float64
 1   armas_medios          1047249 non-null  float64
 2   genero                1047249 non-null  float64
 3   grupo_etario          1047249 non-null  float64
 4   descripción_conducta  1047249 non-null  float64
 5   cantidad              1047249 non-null  int64
dtypes: float64(5), int64(1)
memory usage: 47.9 MB
```

To normalize the data we need to get some values from the data

Data normalization is performed to ensure that all variables are on the same scale. This is done to avoid variables with higher numerical values having a disproportionate weight. The formula for normalization is as follows:

$$x_{norm} = \frac{x - x_{mean}}{std}$$

Where x is the original variable, x_{norm} is the normalized variable, x_{mean} is the minimum value of the variable and std is the standard deviation of the variable.

In [ ]:
```
train_stats = df1.describe()
train_stats
```

Out[ ]:

	departamento	armas_medios	genero	grupo_etario	descripción_conducta	cantidad
count	1.047249e+06	1.047249e+06	1.047249e+06	1.047249e+06	1.047249e+06	1.047249e+06
mean	1.558220e+01	7.022941e+00	6.747555e-01	1.057421e+00	7.147679e-01	1.617188e+00
std	9.722598e+00	4.026562e+00	5.732182e-01	5.896843e-01	4.515251e-01	2.163696e+00
min	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	1.000000e+00
25%	6.000000e+00	4.000000e+00	0.000000e+00	1.000000e+00	0.000000e+00	1.000000e+00
50%	1.500000e+01	5.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00
75%	2.600000e+01	1.300000e+01	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00
max	3.100000e+01	1.400000e+01	2.000000e+00	3.000000e+00	1.000000e+00	1.140000e+02

In [ ]:
```
def norm(x):
  return (x - train_stats.loc['mean']) / train_stats.loc['std']
df2 = norm(df1)
df2 = df.drop(columns='fecha_hecho').to_numpy()
```

In [ ]:
```
df3 = norm(df1)
df3
```

Out[ ]:

	departamento	armas_medios	genero	grupo_etario	descripción_conducta	cantidad
0	-1.499826	-0.502399	-1.177135	-0.097376	0.631708	0.176925
1	-1.499826	-0.502399	0.567401	-0.097376	0.631708	-0.285247
2	-1.499826	-0.502399	0.567401	-0.097376	0.631708	-0.285247
3	-1.499826	-0.502399	-1.177135	-0.097376	0.631708	-0.285247
4	-1.294119	-0.502399	-1.177135	-0.097376	0.631708	0.176925
...	...	...	...	...	...	...
1047244	-0.574147	1.236057	0.567401	-0.097376	0.631708	-0.285247
1047245	0.145825	1.236057	-1.177135	-1.793198	0.631708	-0.285247
1047246	1.277210	1.236057	0.567401	-0.097376	0.631708	-0.285247
1047247	-0.368441	0.739355	0.567401	-0.097376	0.631708	-0.285247
1047248	-0.368441	0.739355	0.567401	-0.097376	0.631708	-0.285247

1047249 rows × 6 columns

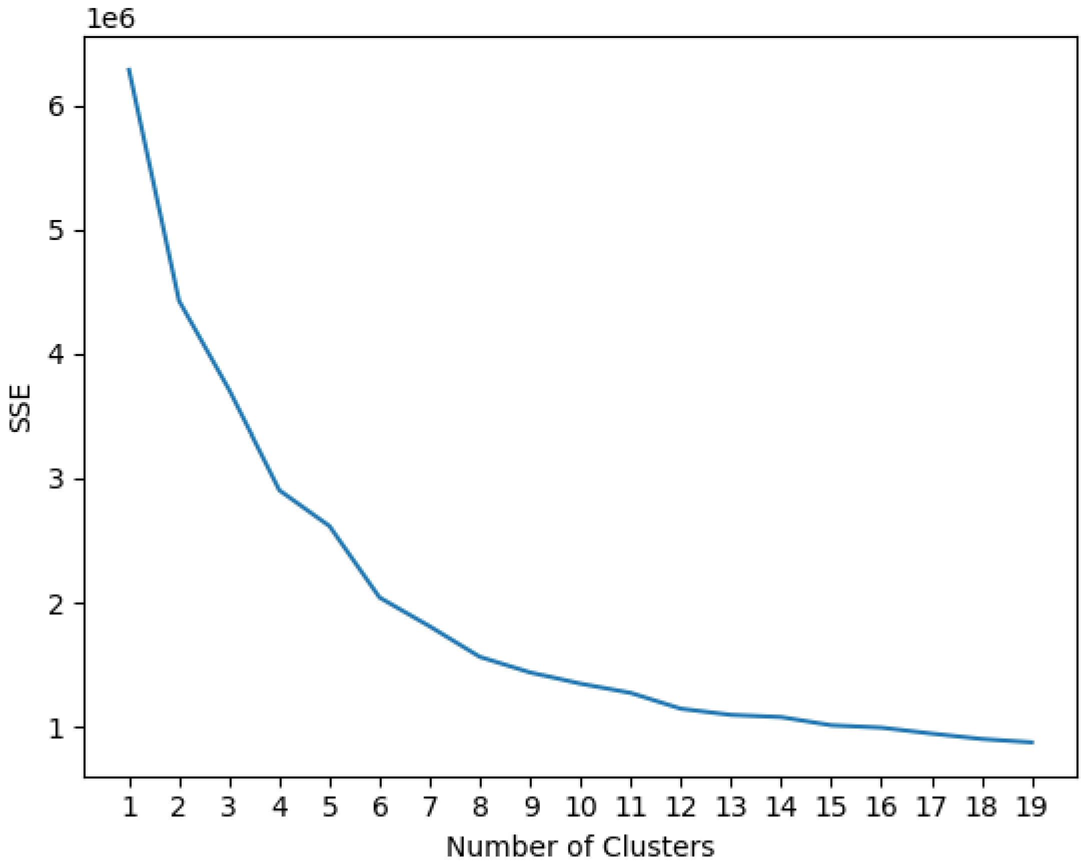
Now, We need to find the K number, Kmeans algorithm needs to number of cluster to create the model, for this reason we use the elbow method to find the K number.

Elbow Method

I'll find th n_cluster that better fit to the data

In [ ]:

```
kmeans_kwargs = {
"init": "random",
"n_init": 10,
"random_state": 1,
}

#create list to hold SSE values for each k
sse = []
for k in range(1, 20):
    kmeans = KMeans(n_clusters=k, **kmeans_kwargs)
    kmeans.fit(df3)
    sse.append(kmeans.inertia_)

#visualize results
plt.plot(range(1, 20), sse)
plt.xticks(range(1, 20))
plt.xlabel("Number of Clusters")
plt.ylabel("SSE")
plt.show()
```



I use Kneed to detect the optimal cluster to build the model, this method allowed me find out the optimal number of cluster gave me as result 6 clusters, Now, I can build the model

In [ ]:
```
from kneed import KneeLocator
cost_knee_c3 = KneeLocator(
    x= range(1,20),
    y=sse,
    S=0.1, curve="convex",
    direction="decreasing", online=True)

K_cost_c3 = cost_knee_c3.elbow
print('Elbow at K =',f'{K_cost_c3:.0f} clusters')
```

```
Elbow at K = 6 clusters
```

Build the model

bluid the model with the cluster that Kneed gave me as result above

In [ ]:
```
# Construir modelo
from sklearn.cluster import KMeans
km = KMeans(init="k-means++", n_clusters=6, max_iter=10000,n_init=20,algorithm='elkan'
km.fit(df3)
```

Out[ ]:
```
KMeans(algorithm='elkan', max_iter=10000, n_clusters=6, n_init=20)
```

In [ ]:
```
km.labels_ #clusters
```

Out[ ]:
```
array([0, 3, 3, ..., 5, 3, 3])
```

In [ ]:
```
km.cluster_centers_ # centroids
```

Out[ ]:
```
array([[-0.58740587, -0.48161008, -1.17713547, -0.239587  ,  0.62796963,
        -0.08132174],
       [ 0.09686951,  1.38980442, -0.06192137, -0.10393527, -1.58057542,
        -0.04963278],
       [-0.07873142, -0.43652053,  2.27684899,  3.29421016,  0.62961823,
        -0.05165761],
       [-0.79556882, -0.52470725,  0.56796301, -0.21311655,  0.62777335,
        -0.09076381],
       [-0.25868609, -0.28856714, -0.02922348,  0.14320751,  0.31397215,
         7.96882627],
       [ 1.03680752, -0.65610897, -0.02149409, -0.24372633,  0.63073792,
        -0.0790793 ]])
```

In [ ]:
```
kmeans.predict(X=df3, sample_weight=5)
```

Out[ ]:
```
array([ 5,  3,  3, ...,  2, 11, 11])
```

In [ ]:
```
# Create the new data frame with cluster
cluster_map = pd.DataFrame()
cluster_map['data_index'] = df1.index.values
cluster_map['cluster'] = km.labels_
```

In [ ]:
```
cluster_map
```

Out[ ]:

	data_index	cluster
0	0	0
1	1	3
2	2	3
3	3	0
4	4	0
...	...	...
1047244	1047244	3
1047245	1047245	0
1047246	1047246	5
1047247	1047247	3
1047248	1047248	3

1047249 rows × 2 columns

In [ ]:
```
groups = pd.concat([df.reset_index(),cluster_map],axis=1) # concatenate this dataframe
groups = groups.drop(columns=['data_index','index'])
groups
```

Out[ ]:

	fecha_hecho	departamento	municipio	armas_medios	genero	grupo_etario	descripción
0	2010-01-01	antioquia	girardota	cortopunzante	femenino	adultos	lesiones
1	2010-01-01	antioquia	girardota	cortopunzante	masculino	adultos	lesiones
2	2010-01-01	antioquia	mutatá	cortopunzante	masculino	adultos	lesiones
3	2010-01-01	antioquia	necoclí	cortopunzante	femenino	adultos	lesiones
4	2010-01-01	atlántico	barranquilla (ct)	cortopunzante	femenino	adultos	lesiones
...	...	...	...	...	...	...	
1047244	2022-05-03	cesar	valledupar (ct)	sustancias tóxicas	masculino	adultos	lesiones
1047245	2022-06-16	huila	oporapa	sustancias tóxicas	femenino	adolescentes	lesiones
1047246	2022-04-17	tolima	ibagué (ct)	sustancias tóxicas	masculino	adultos	lesiones
1047247	2022-03-30	cundinamarca	cota	sin armas	masculino	adultos	lesiones
1047248	2022-06-10	cundinamarca	guaduas	sin armas	masculino	adultos	lesiones

1047249 rows × 11 columns

I notice each result of the cluster with categorical data

In [ ]:

```
groups[groups.cluster == 0].describe(include='object') # cluster 1
```

Out[ ]:

	departamento	municipio	armas_medios	genero	grupo_etario	descripción_conducta
count	197866	197866	197866	197866	197866	197866
unique	25	827	15	1	4	2
top	cundinamarca	bogotá d.c. (ct)	contundentes	femenino	adultos	lesiones personales
freq	36764	17427	104480	197866	171363	197532

In [ ]:

```
groups[groups.cluster == 1].describe(include='object') # cluster 2
```

Out[ ]:

	departamento	municipio	armas_medios	genero	grupo_etario	descripción_conducta
count	296788	296788	296788	296788	296788	296788
unique	32	982	7	3	4	2
top	valle	cali (ct)	vehiculo	masculino	adultos	lesiones culposas
freq	43426	16713	262253	189665	247443	296462

In [ ]:

```
groups[groups.cluster == 2].describe(include='object') # cluster 3
```

Out[ ]:

	departamento	municipio	armas_medios	genero	grupo_etario	descripción_conducta
count	57226	57226	57226	57226	57226	57226
unique	32	993	14	3	2	2
top	cundinamarca	bogotá d.c. (ct)	contundentes	no reporta	no reporta	lesiones personales
freq	9842	3047	31551	56345	57224	57172

In [ ]:

```
groups[groups.cluster == 3].describe(include='object') # cluster 4
```

Out[ ]:

	departamento	municipio	armas_medios	genero	grupo_etario	descripción_conducta
count	223451	223451	223451	223451	223451	223451
unique	21	699	15	2	3	2
top	cundinamarca	bogotá d.c. (ct)	contundentes	masculino	adultos	lesiones personales
freq	49086	22484	94024	223379	196780	223054

In [ ]:

```
groups[groups.cluster == 4].describe(include='object') # cluster 5
```

Out[ ]:

	departamento	municipio	armas_medios	genero	grupo_etario	descripción_conducta
count	9389	9389	9389	9389	9389	9389
unique	27	125	7	3	4	2
top	cundinamarca	bogotá d.c. (ct)	contundentes	masculino	adultos	lesiones personales
freq	7480	7456	3992	4828	8690	8042

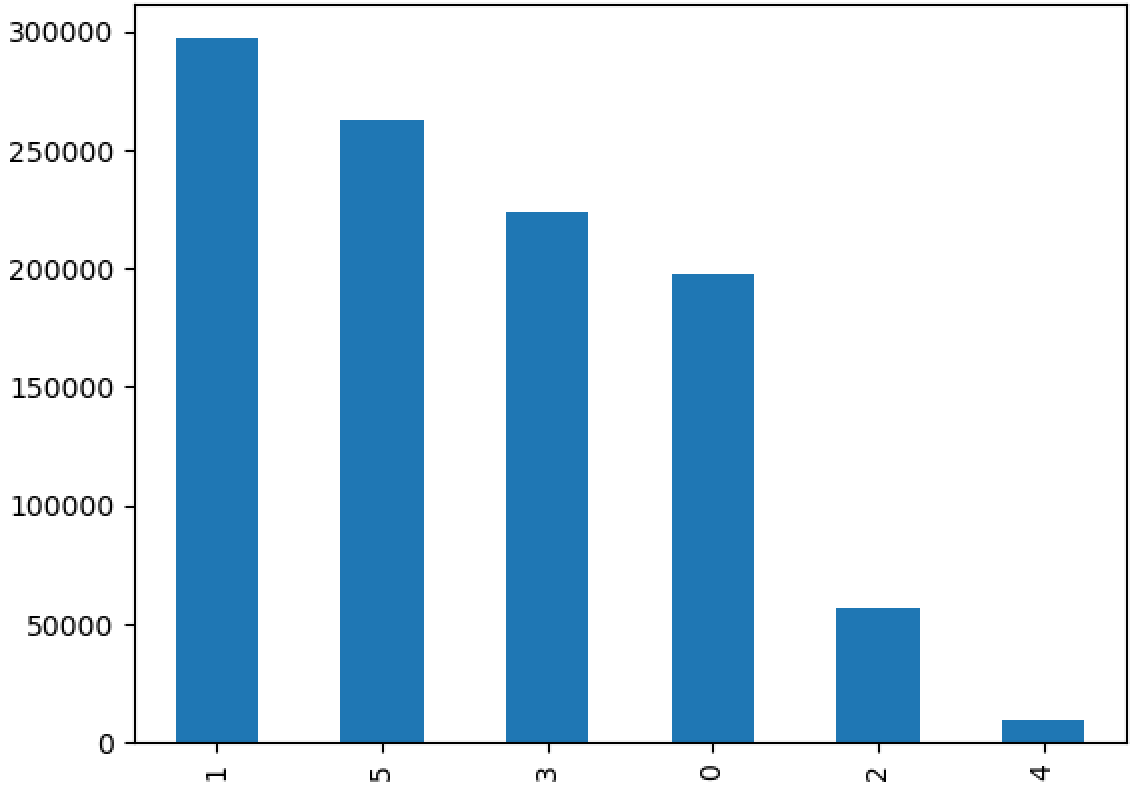
In the plot bellow notice the cluster 4 got less than the other cluster, cluster 1 got more than other clusters.

In [ ]:

```
groups['cluster'].value_counts().plot(kind='bar')
```

Out[ ]:

```
<AxesSubplot:>
```



In [ ]:
```
def denorm(x):
  return (x * train_stats.loc['std'] + train_stats.loc['mean'])
#df4 = df3.drop(columns='cluster')
df4 = denorm(df3)
df4 = pd.concat([df4,cluster_map],axis=1).drop(columns='data_index')
```

In [ ]:
```
df4
```

Out[ ]:

	departamento	armas_medios	genero	grupo_etario	descripción_conducta	cantidad	cluster
0	1.0	5.0	0.0	1.0	1.0	2.0	0
1	1.0	5.0	1.0	1.0	1.0	1.0	3
2	1.0	5.0	1.0	1.0	1.0	1.0	3
3	1.0	5.0	0.0	1.0	1.0	1.0	0
4	3.0	5.0	0.0	1.0	1.0	2.0	0
...	...	...	...	...	...	...	...
1047244	10.0	12.0	1.0	1.0	1.0	1.0	3
1047245	17.0	12.0	0.0	0.0	1.0	1.0	0
1047246	28.0	12.0	1.0	1.0	1.0	1.0	5
1047247	12.0	10.0	1.0	1.0	1.0	1.0	3
1047248	12.0	10.0	1.0	1.0	1.0	1.0	3

1047249 rows × 7 columns

In [ ]:
```
# Inspect the categorical variables
df.select_dtypes('object').nunique()
```

Out[ ]:
```
departamento             32
municipio              1023
armas_medios             15
genero                    3
grupo_etario              4
descripción_conducta      2
dtype: int64
```

In [ ]:
```
# Check missing value
df4.isna().sum()
```

Out[ ]:
```
departamento            0
armas_medios            0
genero                  0
grupo_etario            0
descripción_conducta    0
cantidad                0
cluster                 0
dtype: int64
```

In [ ]:
```
df_region = pd.DataFrame(groups['departamento'].value_counts()).reset_index()
```

In [ ]:
```
df_region['Percentage'] = df_region['departamento'] / groups['departamento'].value_cou
```

In [ ]:
```
df_region.rename(columns = {'index':'departamento', 'departamento':'Total'}, inplace =
df_region
```

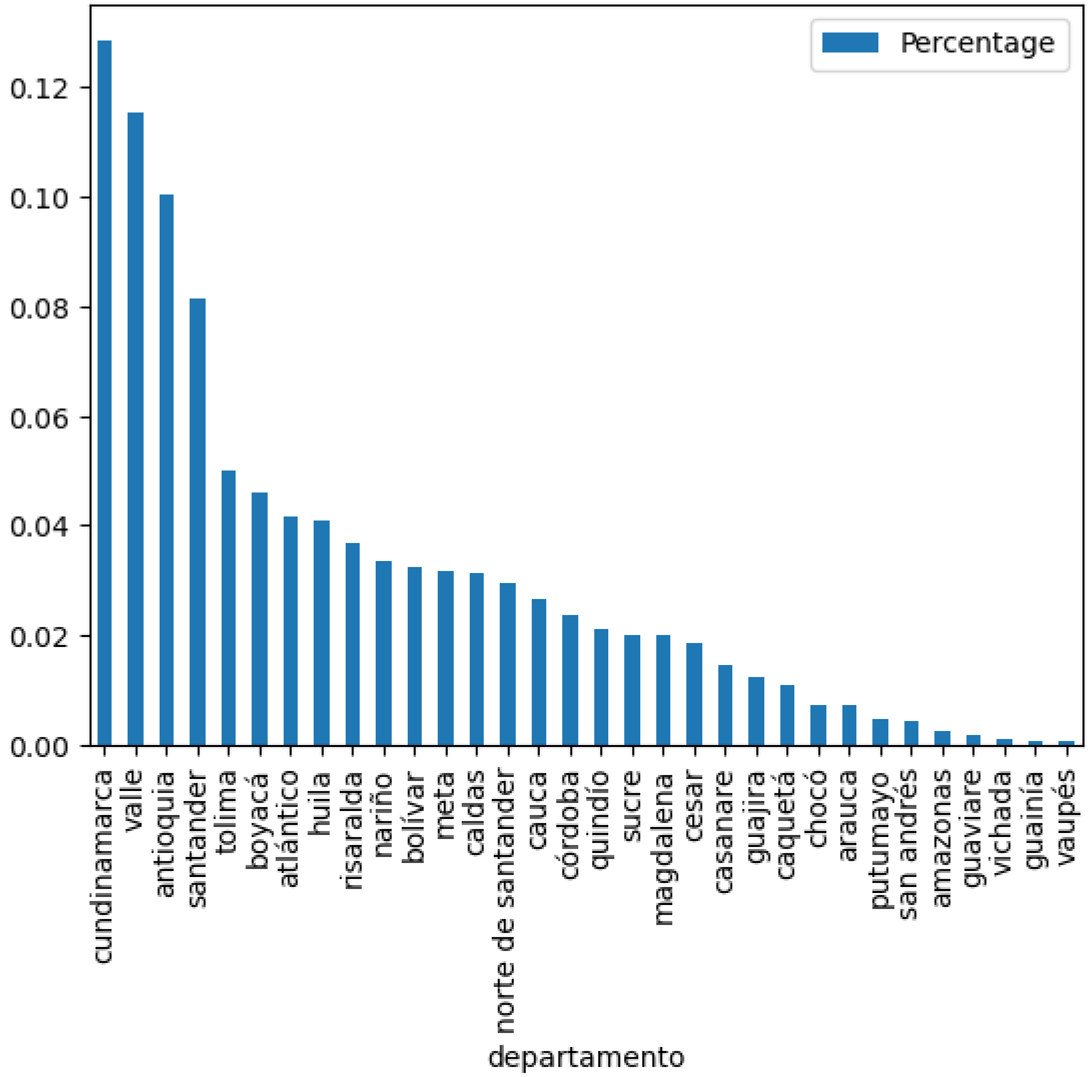
Out[ ]:

	departamento	Total	Percentage
0	cundinamarca	134439	0.128373
1	valle	120891	0.115437
2	antioquia	105105	0.100363
3	santander	85237	0.081391
4	tolima	52423	0.050058
5	boyacá	48113	0.045942
6	atlántico	43755	0.041781
7	huila	42801	0.040870
8	risaralda	38702	0.036956
9	nariño	35336	0.033742
10	bolívar	33979	0.032446
11	meta	33352	0.031847
12	caldas	32739	0.031262
13	norte de santander	30889	0.029495
14	cauca	28050	0.026784
15	córdoba	24939	0.023814
16	quindío	22149	0.021150
17	sucre	21188	0.020232
18	magdalena	21103	0.020151
19	cesar	19384	0.018509
20	casanare	15282	0.014593
21	guajira	13141	0.012548
22	caquetá	11457	0.010940
23	chocó	7716	0.007368
24	arauca	7683	0.007336
25	putumayo	5121	0.004890
26	san andrés	4511	0.004307
27	amazonas	2879	0.002749
28	guaviare	2108	0.002013
29	vichada	1115	0.001065
30	guainía	922	0.000880
31	vaupés	740	0.000707

In [ ]:
```
df_region = df_region.sort_values('Total', ascending = False).reset_index(drop = True)
```

In [ ]:
```
df_region.plot.bar(x='departamento', y='Percentage')
```

Out[ ]: <AxesSubplot:xlabel='departamento'>



This graph shows us the percentage of accidents and personal injuries by department. We can see that Cundinamarca still holds the highest percentage, followed by Antioquia and Valle del Cauca. These departments are the most populous ones in the country, so it is expected that they would have a higher number of accidents and personal injuries. However, it is still concerning to see that the percentage of accidents and personal injuries is quite high in these areas.

It is important to note that some departments, such as Vaupés and Guainía, have very low percentages. These are remote and sparsely populated regions in the country, so it is not surprising that they have lower numbers of accidents and personal injuries.

Overall, this graph gives us an idea of the distribution of accidents and personal injuries by department in Colombia. It can be a useful tool for policymakers and organizations to identify areas where more attention and resources are needed to reduce the incidence of accidents and personal injuries.

In [ ]:
```
# Cluster interpretation
groups.groupby('cluster').agg(
    {
        'departamento': lambda x: x.value_counts().index[0],
        'municipio': lambda x: x.value_counts().index[0],
        'genero': lambda x: x.value_counts().index[0],
        'armas_medios': lambda x: x.value_counts().index[0],
        'grupo_etario': lambda x: x.value_counts().index[0],
        'descripción_conducta': lambda x: x.value_counts().index[0],
        'cantidad': 'mean',
    }
).reset_index()
```

Out[ ]:

	cluster	departamento	municipio	genero	armas_medios	grupo_etario	descripción_conducta	(
0	0	cundinamarca	bogotá d.c. (ct)	femenino	contundentes	adultos	lesiones personales	
1	1	valle	cali (ct)	masculino	vehiculo	adultos	lesiones culposas	
2	2	cundinamarca	bogotá d.c. (ct)	no reporta	contundentes	no reporta	lesiones personales	
3	3	cundinamarca	bogotá d.c. (ct)	masculino	contundentes	adultos	lesiones personales	
4	4	cundinamarca	bogotá d.c. (ct)	masculino	contundentes	adultos	lesiones personales	1
5	5	valle	cali (ct)	masculino	contundentes	adultos	lesiones personales	

In [ ]:
```
Z = groups.copy()
Z = Z.drop(columns=['dia_semana','mes','fecha_hecho','municipio'])
Z = pd.get_dummies(Z)
Z
```

Out[ ]:

	cantidad	departamento_amazonas	departamento_antioquia	departamento_arauca	departam
0	2	0	1	0	
1	1	0	1	0	
2	1	0	1	0	
3	1	0	1	0	
4	2	0	0	0	
...	...	...	...	...	
1047244	1	0	0	0	
1047245	1	0	0	0	
1047246	1	0	0	0	
1047247	1	0	0	0	
1047248	1	0	0	0	

1047249 rows × 63 columns

PCA

In order to vizualize the results of the clusters that we found above, We have use the PCA method to reduce the dimensionality of the data. This algorithm allowed us vizualize the data in 3 dimensions.

In [ ]:

```
from sklearn.decomposition import PCA

# Obtención de componentes principales
pca = PCA(n_components=3)
pca.fit(Z)
transformada=pca.transform(Z)

# Código de visualización

print("Explained Variance for each component:", pca.explained_variance_)
print("Explainded Variance Ratio for each component:", pca.explained_variance_ratio_)
```

```
Varianza explicada por cada componente: [4.6938254  0.91859283 0.56984193]
Proporción de varianza explicada por cada componente: [0.55876213 0.10935108 0.067835
09]
```

In [ ]:

```
dict_cluster = {0:'c1',1:'c2',2:'c3',3:'c4',4:'c5',5:'c6'}
```

In [ ]:

```
groups['cluster'] = groups['cluster'].replace(dict_cluster)
groups
```

Out[ ]:

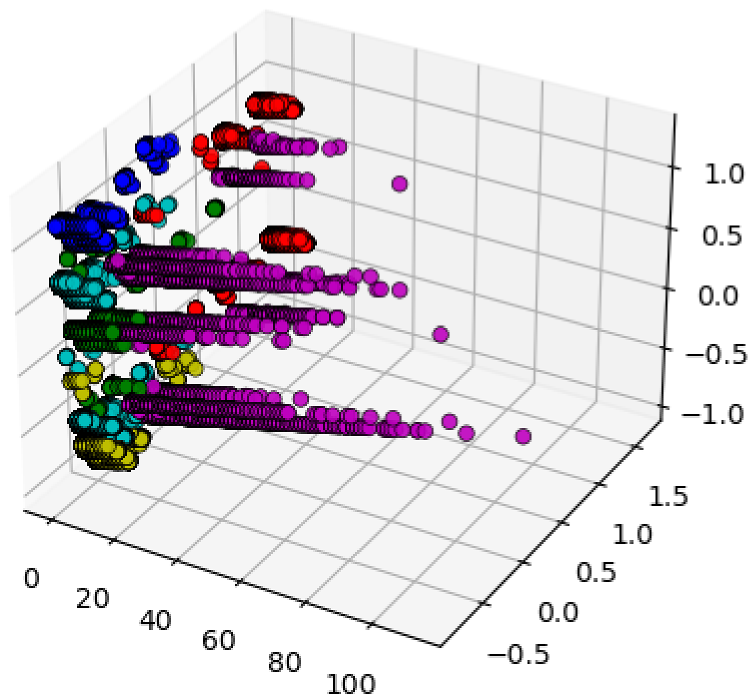
	fecha_hecho	departamento	municipio	armas_medios	genero	grupo_etario	descripción
0	2010-01-01	antioquia	girardota	cortopunzante	femenino	adultos	lesiones
1	2010-01-01	antioquia	girardota	cortopunzante	masculino	adultos	lesiones
2	2010-01-01	antioquia	mutatá	cortopunzante	masculino	adultos	lesiones
3	2010-01-01	antioquia	necoclí	cortopunzante	femenino	adultos	lesiones
4	2010-01-01	atlántico	barranquilla (ct)	cortopunzante	femenino	adultos	lesiones
...	...	...	...	...	...	...	
1047244	2022-05-03	cesar	valledupar (ct)	sustancias tóxicas	masculino	adultos	lesiones
1047245	2022-06-16	huila	oporapa	sustancias tóxicas	femenino	adolescentes	lesiones
1047246	2022-04-17	tolima	ibagué (ct)	sustancias tóxicas	masculino	adultos	lesiones
1047247	2022-03-30	cundinamarca	cota	sin armas	masculino	adultos	lesiones
1047248	2022-06-10	cundinamarca	guaduas	sin armas	masculino	adultos	lesiones

1047249 rows × 11 columns

In [ ]:

```
# Scatter

# Import libraries
from mpl_toolkits.mplot3d import axes3d
import matplotlib.pyplot as plt

# create figure
fig = plt.figure()
# Create 3D
ax1 = fig.add_subplot(111, projection='3d')

# Defining the data
x = transformada[:,0]
y = transformada[:,1]
z = transformada[:,2]

# Defining the colors

#color = df['genero'].map({'masculino':'b','femenino':'r','no reporta':'g'})

color = groups['cluster'].map({'c1':'b', 'c2':'r', 'c3':'g', 'c4':'y', 'c5':'m', 'c6':

# make the scatter plot
```

```
ax1.scatter(x, y, z, s=30, c=color, alpha=0.9, edgecolors='k', linewidths=0.5)

# show the plot

plt.show()
```



This is the result of the clustering algorithm applied to the data. The graph shows how the algorithm grouped the data and how it is distributed in 3 dimensions. We can observe that there are clear clusters with a significant amount of data points that are tightly packed together, while other points seem to be more spread out. The clustering algorithm can be a useful tool to identify patterns and groupings in data that might not be immediately apparent, providing insights and aiding decision-making processes. However, it is important to keep in mind that the results of the clustering algorithm are only as good as the data and the chosen parameters, and may require further analysis and refinement.

Conclusions

The analysis of accidents in Colombia has allowed us to delve deeper into the behavior of accidents, discover how they affect different ages and genders, and analyze the type of weapons involved. Machine learning algorithms have helped us to identify patterns and trends in accidents, and provide decision-makers with important data to help create effective programs and policies to improve safety throughout the country. This initiative offers an excellent opportunity to contribute to Colombia's safety and improve the lives of its citizens. Our team is committed to working on this initiative to ensure that this important task is carried out efficiently and effectively. We are convinced that our work will make a significant contribution to improving the security and quality of life of Colombians.

Based on the information and data presented, some conclusions that could be drawn from this project are:

- Blunt and sharp objects, as well as vehicular accidents, are the most common causes of personal injuries and accidents in Colombia.
- The majority of these accidents seem to be caused by unintentional events, although intentional acts of violence cannot be ruled out. Men, particularly adult men, are more likely to be affected by personal injuries and accidents than women.
- The use of blunt objects seems to affect women more than sharp objects.
- Violence in the streets and intrafamily violence could be important factors contributing to personal injuries and accidents.
- There is a need for further research and data analysis to better understand the causes and circumstances surrounding personal injuries and accidents in Colombia.
- These findings suggest the importance of implementing policies and measures aimed at preventing accidents and reducing violence in the country.

Additionally, it highlights the need for more comprehensive data collection and analysis to better understand the causes of injuries and accidents, particularly those related to violence and criminal activity. Policymakers and public health officials can use this information to develop targeted interventions and preventive measures to reduce the incidence of injuries and improve the overall health and safety of the population.

Overall, this project underscores the importance of data-driven approaches to public health and safety, as well as the potential of data visualization tools to communicate complex information in an accessible and actionable way.

I invite you to take a look at the dashboard I designed on Power BI, where you can explore and analyze the data of accidents and personal injuries in Colombia. You will find several interactive visualizations that will allow you to dig deeper into the information and understand the patterns and trends of this problem in the country. To access the dashboard, please follow this link: https://onx.la/71e87. I hope you find it interesting and informative. Let me know if you have any questions or feedback!

Some Suggestions

Include more data: The dataset used in this project is limited to the years 2014-2018 and to reported cases only. To obtain a more comprehensive understanding of the issue, it would be helpful to gather data from a wider time frame and include unreported cases as well.

Include more variables: While the current dataset provides valuable information on the type of accidents and injuries that occur in Colombia, including more variables such as the location and time of day of the incidents could provide further insights into the issue.

Further analysis: The current project provides a good overview of the trends and patterns of accidents and injuries in Colombia. However, conducting further analysis using advanced statistical techniques could uncover more complex relationships between variables and provide a more nuanced understanding of the issue.

Collaboration with local authorities: To address the issue of accidents and injuries in Colombia, it would be helpful for researchers to collaborate with local authorities to develop and implement targeted interventions and policies aimed at preventing accidents and injuries.

Future approaches

- Conducting a more in-depth analysis of the causes and circumstances behind the injuries and accidents in each department, as well as their distribution by gender and age. This could provide more insights into the root causes of the injuries and help identify potential interventions.
- Examining the economic costs associated with injuries and accidents, including medical expenses, lost income, and disability costs. This could help policymakers prioritize interventions and allocate resources more effectively.
- Studying the effectiveness of existing policies and interventions aimed at reducing injuries and accidents, and identifying areas for improvement. This could involve evaluating specific policies, such as traffic safety laws or regulations on the use of weapons, and analyzing their impact on injury rates.
- Using machine learning algorithms to predict injury rates in different regions of the country based on demographic, economic, and social indicators. This could help identify regions at risk and target interventions more effectively.
- Collaborating with local communities and organizations to develop tailored interventions that address specific needs and challenges. This could involve working with community leaders, healthcare providers, and government agencies to develop and implement evidence-based programs and policies.