



Análise de Dados

Grupo 2

24 de janeiro de 2021



Angélica Freitas
A83761



António Lindo
A85813



Beatriz Rocha
A84003



Rodrigo Pimentel
A83765

Conteúdo

1	Introdução	4
2	Fonte de Dados	5
3	Arquitetura	6
3.1	Método dos 4 passos	6
3.1.1	Passo 1 - Selecionar o processo de negócio	6
3.1.2	Passo 2 - Definir a granularidade	6
3.1.3	Passo 3 - Identificar as dimensões	6
3.1.4	Passo 4 - Identificar os factos	7
3.2	Criar <i>schema</i>	8
4	<i>ETL - Extract, Transform, Load</i>	9
4.1	Extração	16
4.2	Transformação	16
4.3	Carregamento	18
5	Indicadores de <i>Business Intelligence</i>	21
6	Conclusão	28

Lista de Figuras

3.1	Arquitetura seguida	6
3.2	Tabela de factos e respectivas dimensões	8
3.3	Modelo em floco de neve	8
4.1	<i>Dataset</i> original	9
4.2	Seleção da primeira coluna	10
4.3	Text to Columns 1	11
4.4	Text to Columns 2	12
4.5	<i>Dataset</i> após a aplicação da ferramenta Text to Columns	12
4.6	Formatar data no <i>dataset</i>	13
4.7	<i>Dataset</i> final	14
5.1	Número total de incidentes por estado	21
5.2	Número de suspeitos por faixa etária	22
5.3	Número de presos por estado	22
5.4	Número total de incidentes por ano e mês	23
5.5	Número de incidentes com cada tipo de arma	23
5.6	Número de suspeitos por relação e género	24
5.7	Média de mortes por incidente por estado	24
5.8	Número de mortos e feridos por género	25
5.9	Número de suspeitos presos, feridos e mortos	25
5.10	Número de vítimas por género	26
5.11	Média de idades de suspeitos que roubaram e não roubaram armas	27

Lista de Tabelas

2.1	Descrição dos atributos do <i>dataset</i>	5
3.1	dim_gun	7
3.2	dim_gun_type	7
3.3	dim_gun_stolen	7
3.4	dim_date	7
3.5	dim_participant	7
3.6	dim_participant_age_group	7
3.7	dim_incident_info	7
3.8	dim_location	7
3.9	dim_state_district	7

1 Introdução

Nas últimas décadas, temos vindo a observar um crescente aumento de fontes de dados e, em contrapartida, uma redução dos custos de armazenamento dos dados resultantes, o que tem vindo a provocar uma grande necessidade de os analisar. Atualmente, a maior parte das empresas recorre a ferramentas de suporte à decisão que oferecem a possibilidade de tirar conclusões acerca dos dados das mesmas e, consequentemente, a oportunidade de tomar decisões de forma a que possam manter a competitividade. Contudo, para isso, estas necessitam de armazenar a informação de forma consolidada e é aí que surgem os *data warehouses* que, por sua vez, integram os dados internos e externos das mesmas numa única estrutura, permitindo uma melhor utilização da informação e aumento da capacidade de resposta e adaptação.

Deste modo, o objetivo deste trabalho prático passa por analisar, planear e implementar um sistema de bases de dados multidimensionais. Para a sua realização, iremos basear-nos num *dataset* contendo incidentes de violência à mão armada entre 2013 e 2018 nos Estados Unidos da América.

Resumidamente, será feita a projeção e implementação de um *data warehouse* para armazenar os dados mencionados anteriormente. Em primeiro lugar, será criada a base de dados para albergar os mesmos, de seguida será feito o seu povoamento inicial (recorrendo, para isso, a processos de *ETL* e à linguagem de programação Python) e, por último, serão criados alguns indicadores de *Business Intelligence* relevantes para o caso de estudo (recorrendo, para isso, à ferramenta de visualização Tableau).

2 Fonte de Dados

O primeiro passo para realizar este trabalho prático consistiu na escolha de um *dataset* acerca de uma área de negócio em específico. No nosso caso, optámos por escolher a área de criminalidade, sendo que o conjunto de dados escolhido envolve os casos de violência à mão armada entre 2013 e 2018 nos Estados Unidos da América. É importante mencionar que, na escolha do *dataset*, tivemos em conta uma série de requisitos que a seguir se apresentam:

- O *dataset* deve conter datas, uma vez que, num *data warehouse*, os dados estão relacionados com um determinado período de tempo, revelando informações de um ponto de vista histórico;
- O *dataset* não deve estar pré-processado, uma vez que um dos objetivos do trabalho prático é manipular os dados com processos de *ETL*;
- O *dataset* deverá conter, pelo menos, 5000 registos para que a experiência se assemelhe àquilo que acontece no mundo do trabalho o mais possível;
- O *dataset* deverá apresentar várias colunas, ou seja, vários atributos de modo a que seja possível criar vários e bons indicadores de *Business Intelligence*.

Na Tabela 2.1, pode ser vista uma descrição de todos os atributos deste *dataset* para uma melhor compreensão dos mesmos.

Tabela 2.1: Descrição dos atributos do *dataset*

Atributos	Descrição
incident_id	Identificador do crime
date	Data do crime
state	Estado onde ocorreu o crime
city_or_county	Cidade ou concelho onde ocorreu o crime
address	Endereço do local onde ocorreu o crime
n_killed	Número de pessoas mortas
n_injured	Número de pessoas feridas
incident_url	URL relativo ao crime
source_url	URL da fonte informativa
incident_url_fields_missing	TRUE se o 8.º atributo está presente, FALSE caso contrário
congressional_district	Distrito congressional
gun_stolen	Estado das armas envolvidas no crime
gun_type	Tipo das armas envolvidas no crime
incident_characteristics	Características do crime
latitude	Local do crime (latitude)
location_description	Descrição do local onde ocorreu o crime
longitude	Local do crime (longitude)
n_guns_involved	Número de armas envolvidas no crime
notes	Informação adicional sobre o crime
participant_age	Idade do(s) participante(s) do crime
participant_age_group	Faixa etária do(s) participante(s) do crime
participant_gender	Género do(s) participante(s)
participant_name	Nome do(s) participante(s) envolvido(s) no crime
participant_relationship	Relação do participante com outro(s) participante(s)
participant_status	Danos causados ao participante
participant_type	Tipo do participante
sources	Fonte dos participantes
state_house_district	Distrito eleitoral
state_senate_district	Distrito territorial onde um senador estadual é eleito

3 Arquitetura

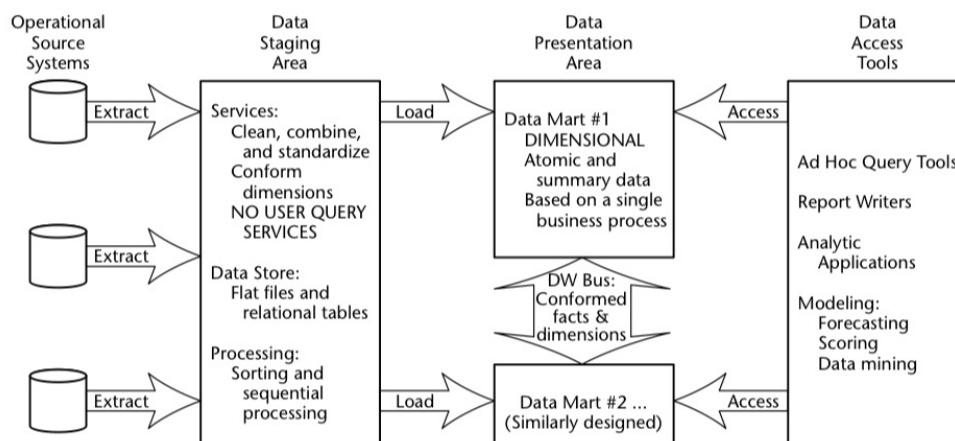


Figura 3.1: Arquitetura seguida

3.1 Método dos 4 passos

Para o desenvolvimento do esquema dimensional, seguimos o método dos 4 passos de Kimball, que pressupõe uma abordagem *bottom-up* para o desenvolvimento do sistema de *data warehousing* e que passamos a explicar nas secções seguintes.

3.1.1 Passo 1 - Selecionar o processo de negócio

O primeiro passo da criação de um *data warehouse* deve ter como foco o processo de negócio mais crítico e útil para o utilizador do primeiro. No nosso caso de estudo, o processo de negócio a modelar será os incidentes, ou seja, os crimes que ocorreram. Estes dados possibilitarão que os utilizadores do *data warehouse* lhe façam interrogações que permitam analisar pormenores como o estado em que houve mais crimes, os tipos de arma mais usados, entre outros.

3.1.2 Passo 2 - Definir a granularidade

No contexto dos incidentes de criminalidade, é preferível escolher o nível mais atómico possível para os registos da tabela de factos. Esta escolha, por sua vez, fornece a maior flexibilidade analítica possível, ou seja, as consultas ao *data warehouse* são muito precisas e os seus dados podem ser limitados e acumulados livremente.

3.1.3 Passo 3 - Identificar as dimensões

Definida a granularidade, torna-se fácil escolher quais as dimensões a analisar. São elas **gun_stolen** (estado da arma), **gun_type** (tipo da arma), **gun** (arma), **date** (data), **participant** (participante), **participant_age_group** (faixa etária do participante), **location** (localização), **state_district** (distrito do incidente) e **incident_info** (informação do incidente).

Dimensão Gun
Gun Key
Incident Key (FK)
Gun Stolen Key (FK)
Gun Type Key (FK)

Tabela 3.1: dim_gun

Dimensão Gun Type
Gun Type Key
Class Type

Tabela 3.2: dim_gun_type

Dimensão Gun Stolen
Gun Stolen Key
Class Stolen

Tabela 3.3: dim_gun_stolen

Dimensão Date
Date Key
Date
Day
Month
Year

Tabela 3.4: dim_date

Dimensão Participant
Participant Key
Gender
Name
Relationship
Status
Type
Participant Age Group Key (FK)
Age
Incident Key (FK)

Tabela 3.5: dim_participant

Dimensão Participant Age Group
Age Group Key
Class Age Group

Tabela 3.6: dim_participant_age_group

Dimensão Incident Info
Incident Info Key
Incident Characteristics
Notes

Tabela 3.7: dim_incident_info

Dimensão Location
Location Key
City or County
State
Longitude
Latitude
Address
Location Description
State District Key (FK)

Tabela 3.8: dim_location

Dimensão State district
State District Key
Senate
House

Tabela 3.9: dim_state_district

3.1.4 Passo 4 - Identificar os factos

Por último, resta apenas identificar os factos que farão parte da tabela de factos que pode ser vista na Figura 3.2.

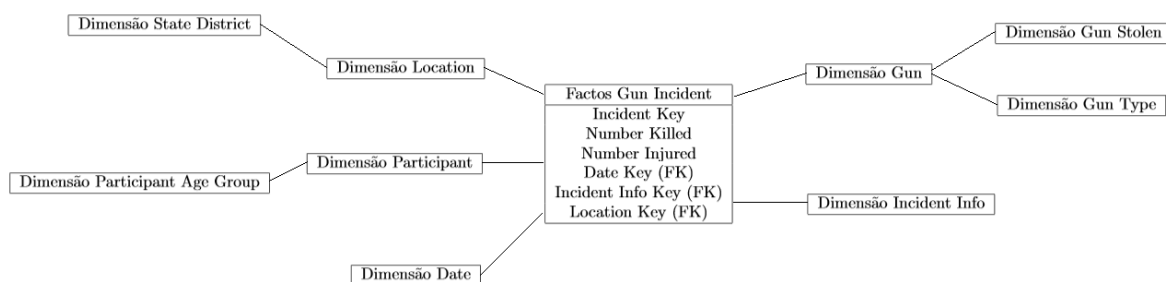


Figura 3.2: Tabela de factos e respetivas dimensões

3.2 Criar *schema*

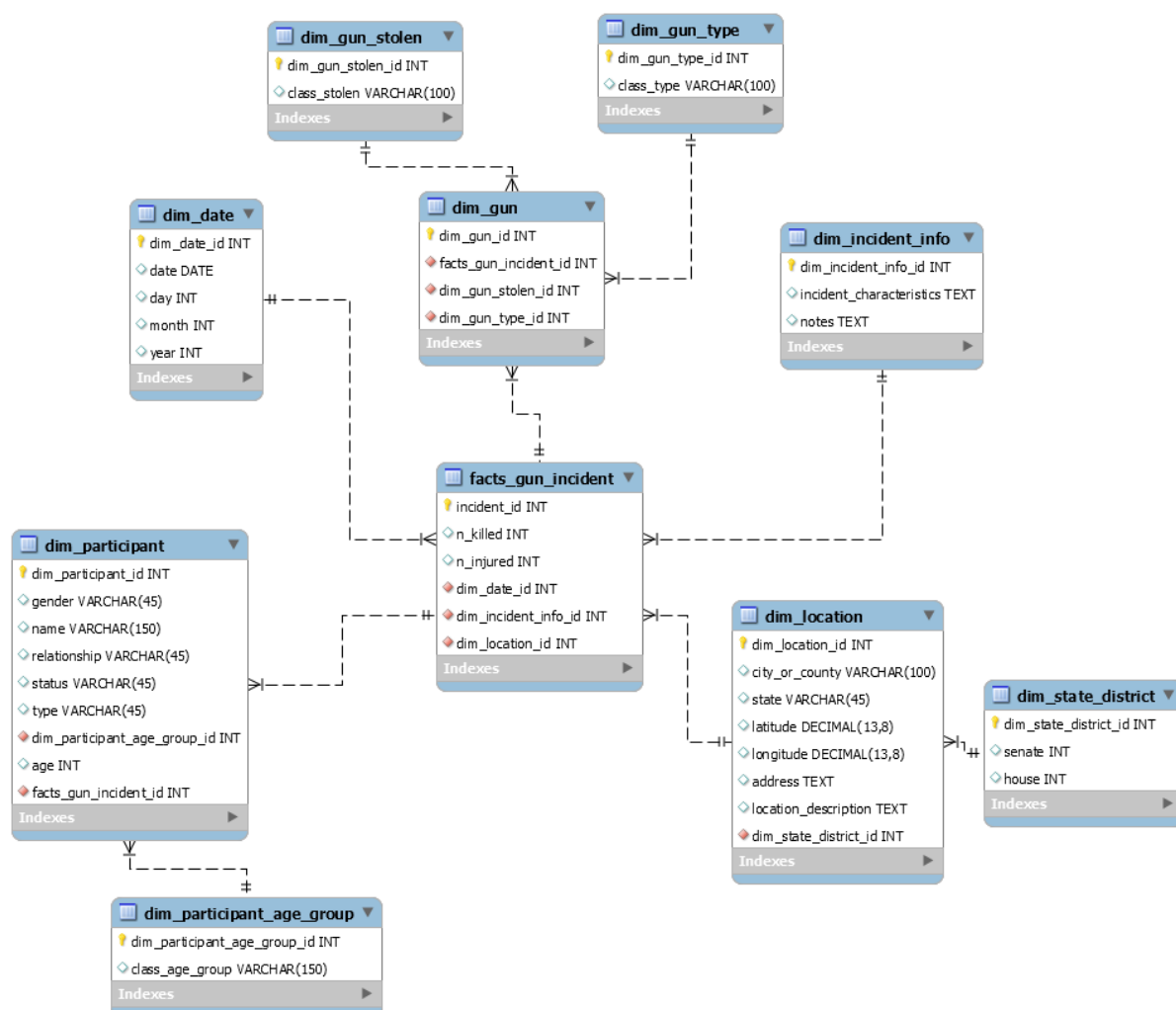


Figura 3.3: Modelo em floco de neve

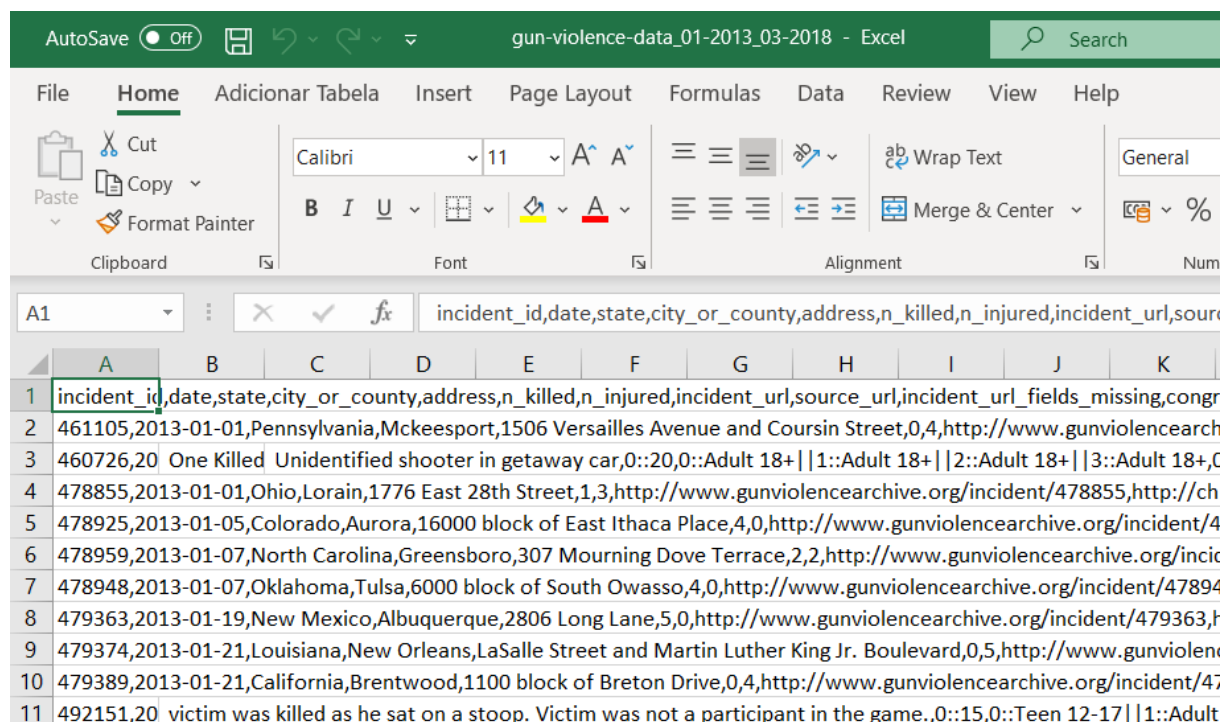
Findo o método dos 4 passos, podemos agora criar o *schema* adequado ao caso de estudo. Optámos por criar um modelo em floco de neve, devido ao facto de ocupar menos espaço em disco e uma vez que se tornava necessário implementar um maior número de tabelas de dimensão, visto que se verificavam vários níveis de relação.

4 *ETL - Extract, Transform, Load*

O processo de *ETL* consiste na extração de dados, seguida da sua transformação e, por fim, do seu carregamento. Para este projeto, recorremos à linguagem de programação Python para realizar este mesmo processo, pois pensamos que esta linguagem é bastante acessível e, assim, temos um maior controlo sobre os dados. Também utilizámos a ferramenta MySQL Workbench para visualização e carregamento dos dados.

Primeiramente, exportámos o *dataset* original para o formato Excel. Sendo que o formato original do ficheiro era *CSV*, qualquer mudança realizada no mesmo não seria guardada, daí a necessidade de mudar.

De seguida, como todos os campos estavam na primeira célula de cada linha, recorremos à ferramenta **Text to Columns** do Excel que pode ser encontrada na secção **Data Tools** da secção **Data**.



	A	B	C	D	E	F	G	H	I	J	K
1	incident_id,date,state,city_or_county,address,n_killed,n_injured,incident_url,source_url,incident_url_fields_missing,congr										
2	461105,2013-01-01,Pennsylvania,Mckeesport,1506 Versailles Avenue and Coursin Street,0,4,http://www.gunviolencearch										
3	460726,20 One Killed Unidentified shooter in getaway car,0::20,0::Adult 18+ 1::Adult 18+ 2::Adult 18+ 3::Adult 18+,C										
4	478855,2013-01-01,Ohio,Lorain,1776 East 28th Street,1,3,http://www.gunviolencearchive.org/incident/478855,http://ch										
5	478925,2013-01-05,Colorado,Aurora,16000 block of East Ithaca Place,4,0,http://www.gunviolencearchive.org/incident/4										
6	478959,2013-01-07,North Carolina,Greensboro,307 Mourning Dove Terrace,2,2,http://www.gunviolencearchive.org/incic										
7	478948,2013-01-07,Oklahoma,Tulsa,6000 block of South Owasso,4,0,http://www.gunviolencearchive.org/incident/47894										
8	479363,2013-01-19,New Mexico,Albuquerque,2806 Long Lane,5,0,http://www.gunviolencearchive.org/incident/479363,t										
9	479374,2013-01-21,Louisiana,New Orleans,LaSalle Street and Martin Luther King Jr. Boulevard,0,5,http://www.gunviolenc										
10	479389,2013-01-21,California,Brentwood,1100 block of Breton Drive,0,4,http://www.gunviolencearchive.org/incident/47										
11	492151,20 victim was killed as he sat on a stoop. Victim was not a participant in the game.,0::15,0::Teen 12-17 1::Adult										

Figura 4.1: *Dataset* original

AutoSave Off		gun-violence-data_01-2013_03-2018 - Excel													
File		Home		Adicionar Tabela		Insert		Page Layout		Formulas		Data		Review	
<div><div><div>Cut</div><div>Copy</div><div>Paste</div><div>Format Painter</div></div><div>Clipboard</div></div>		<div><div><div>Calibri</div><div>11</div><div>A[^]</div><div>A^v</div></div><div><div><div>B</div><div>I</div><div>U</div><div></div><div></div><div></div><div></div></div><div>Font</div></div></div>		<div><div><div></div><div></div><div></div><div></div><div></div></div><div>Alignment</div></div>											
A1															
A		B		C		D		E		F		G		H	
1		incident_id,date,state,city_or_county,address,n_killed,n_injured,incident_url,source_url,inci													
2		461105,2013-01-01,Pennsylvania,Mckeesport,1506 Versailles Avenue and Coursin Street,0,4,f													
3		460726,20 One Killed Unidentified shooter in getaway car,0::20,0::Adult 18+ 1::Adult 18+													
4		478855,2013-01-01,Ohio,Lorain,1776 East 28th Street,1,3,http://www.gunviolencearchive.or													
5		478925,2013-01-05,Colorado,Aurora,16000 block of East Ithaca Place,4,0,http://www.gunvic													
6		478959,2013-01-07,North Carolina,Greensboro,307 Mourning Dove Terrace,2,2,http://www.													
7		478948,2013-01-07,Oklahoma,Tulsa,6000 block of South Owasso,4,0,http://www.gunviolenc													
8		479363,2013-01-19,New Mexico,Albuquerque,2806 Long Lane,5,0,http://www.gunviolencear													
9		479374,2013-01-21,Louisiana,New Orleans,LaSalle Street and Martin Luther King Jr. Boulevar													
10		479389,2013-01-21,California,Brentwood,1100 block of Breton Drive,0,4,http://www.gunviol													
11		492151,20 victim was killed as he sat on a stoop. Victim was not a participant in the game.,0													
12		491674,20 3 inured shooting on Dodds Ave.													
13															
14		35.022083, -85.269986",0::19,0::Adult 18+,0::Male 1::Male 2::Male 3::Male,0::Demetriu													
15		479413,2013-01-25,Missouri,Saint Louis,W Florissant Ave and Riverview Blvd,1,3,http://www													
16		479561,20 deputies hailed from St. Mary Parish sheriff's office. Lyons shot before arson.,3::7													
17		479554,2013-01-26,District of Columbia,Washington,2403 Benning Road Northeast,0,5,http:/													
18		479460,2013-01-26,Ohio,Springfield,601 West Main Street,1,3,http://www.gunviolencearchiv													
19		479572,2013-02-02,Tennessee,Memphis,2514 Mount Meriah,0,5,http://www.gunviolencear													

Figura 4.2: Seleção da primeira coluna

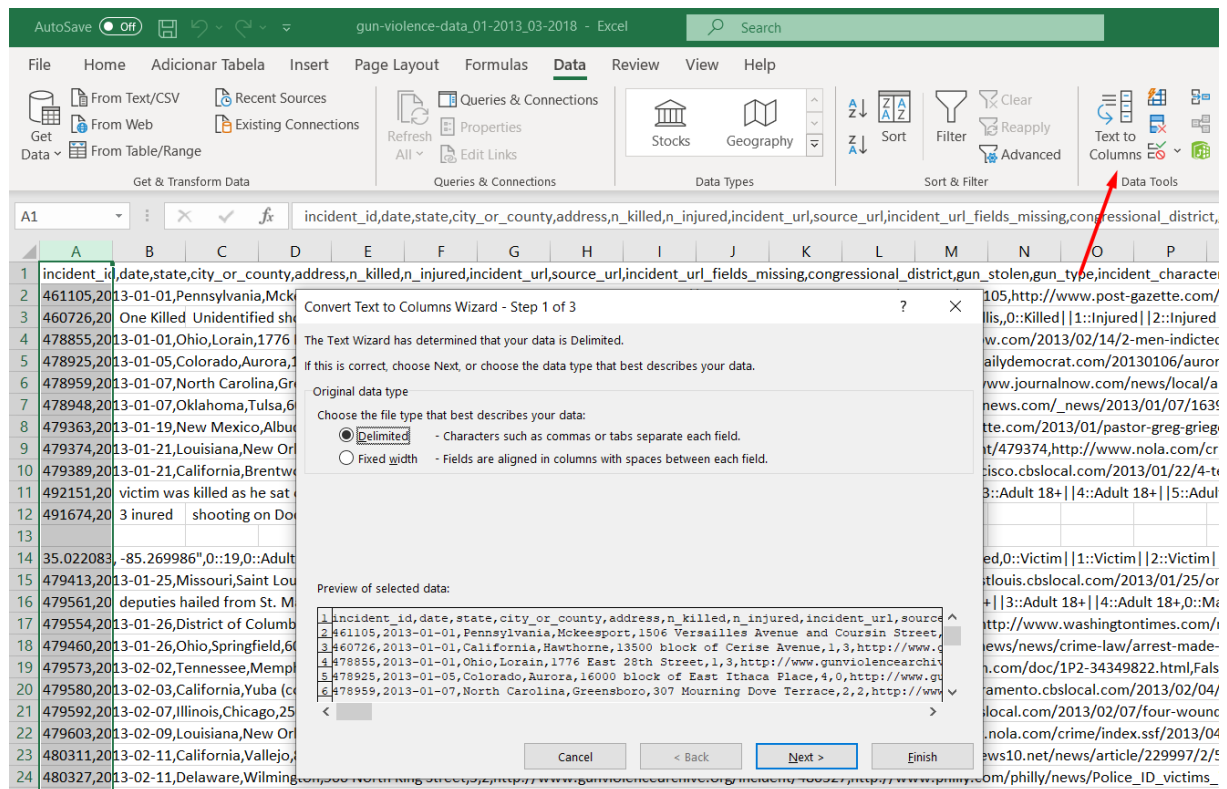


Figura 4.3: Text to Columns 1

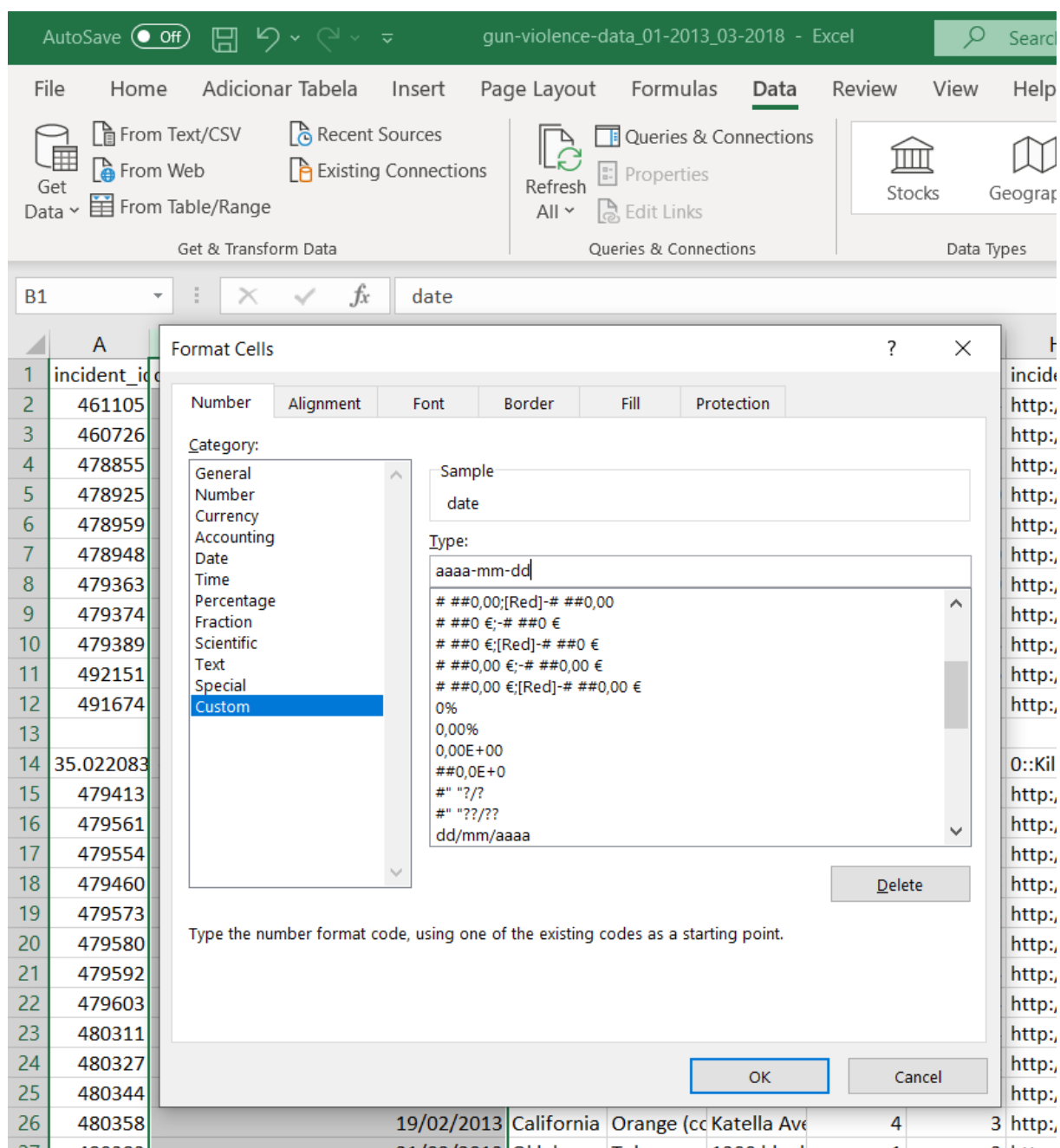


Figura 4.6: Formatar data no *dataset*

E, finalmente, obtivemos o *dataset* já tratado e pronto para ser manipulado em Python.

gun-violence-data_01-2013_03-2018 - Excel																
incident_id																
incident_id	date	state	city_or_co	address	n_killed	n_injured	incident_u	source_url	incident_u	congressio	gun_stoler	gun_type	incident_c	latitude	location	
461105	2013-01-01	Pennsylv	McKeespo	1506 Vers	0	4	http://ww	http://ww	FALSE	14			Shot - Woi	40.3467		
460726	2013-01-01	California	Hawthorn	13500 blo	1	3	http://ww	http://ww	FALSE	43			Shot - Woi	33.909		
478855	2013-01-01	Ohio	Lorain	1776 East	1	3	http://ww	http://chr	FALSE	9	0::Unknown	0::Unknown	Shot - Woi	41.4455	Cotton Cl	
478925	2013-01-05	Colorado	Aurora	16000 blo	4	0	http://ww	http://ww	FALSE	6			Shot - Dea	39.6518		
478959	2013-01-07	North Car	Greensbor	307 Mourr	2	2	http://ww	http://ww	FALSE	6	0::Unknown	0::Handgu	Shot - Woi	36.114		
478948	2013-01-07	Oklahoma	Tulsa	6000 blo	4	0	http://ww	http://usn	FALSE	1			Shot - Dea	36.2405	Fairmont	
479363	2013-01-19	New Mexi	Albuquerq	2806 Long	5	0	http://ww	http://hint	FALSE	1	0::Unknown	0::22 LR	Shot - Dea	34.9791		
479374	2013-01-21	Louisiana	New Orlea	LaSalle Str	0	5	http://ww	http://ww	FALSE	2			Shot - Woi	29.9435		
479389	2013-01-21	California	Brentwoor	1100 blo	0	4	http://ww	http://san	FALSE	9			Shot - Woi	37.9656		
492151	2013-01-23	Maryland	Baltimore	1500 blo	1	6	http://ww	http://ww	FALSE	7			Shot - Woi	39.2899		
491674	2013-01-23	Tennessee	Chattano	1501 Dodc	1	3	http://ww	http://ww	FALSE	3	0::Unknown	0::Unknown	Shot - Woi	35.0221		
35.022083	-85.269986"	0::19	0::Adult	18 0::Male	0::Demetrius Davis	0::Killed	0::Victim	http://ww		28	10					
479413	2013-01-25	Missouri	Saint Louis	W Florissa	1	3	http://ww	http://stlo	FALSE	1	0::Unknown	0::Unknown	Shot - Woi	38.7067		
479561	2013-01-26	Louisiana	Charenton	1000 blo	2	3	http://ww	http://ww	FALSE	3	0::Unknown	0::Shotgun	Shot - Woi	29.8816		
479554	2013-01-26	District of	Washingto	2403 Benn	0	5	http://ww	http://ww	FALSE	1	0::Unknown	0::Handgu	Shot - Woi	38.8978		
479460	2013-01-26	Ohio	Springfield	601 West	1	3	http://ww	http://ww	FALSE	8			Shot - Woi	39.9252	Nite Owl	
479573	2013-02-02	Tennessee	Memphis	2514 Mou	0	5	http://ww	https://ww	FALSE	9	0::Unknown	0::Handgu	Shot - Woi	35.0803	Club Veni	
479580	2013-02-03	California	Yuba (cour	5800 blo	1	3	http://ww	http://sac	FALSE	3	0::Unknown	0::9mm	Shot - Woi	39.1236		
479592	2013-02-07	Illinois	Chicago	2500 blo	0	4	http://ww	http://chic	FALSE	2			Shot - Woi	41.7592		
479603	2013-02-09	Louisiana	New Orlea	400 blo	0	4	http://ww	http://ww	FALSE	2	0::Unknown	0::Handgu	Shot - Woi	29.9563		
480311	2013-02-11	California	Vallejo	800 blo	1	4	http://ww	http://arc	FALSE	5			Shot - Woi	38.1072		
480327	2013-02-11	Delaware	Wilmington	500 North	3	2	http://ww	http://ww	FALSE	1	0::Unknown	0::45 Auto	Shot - Woi	39.7407	New Cast	
480344	2013-02-12	Utah	Midvale	8286 Adan	4	1	http://www	gunviole	FALSE	4			Shot - Woi	40.6008		
480358	2013-02-19	California	Orange (cc	Katella Ave	4	3	http://ww	http://ww	FALSE	46	0::Unknown	0::12 gaug	Shot - Woi	33.8031		
480383	2013-02-21	Oklahoma	Tulsa	1200 blo	1	3	http://ww	http://ww	FALSE	1			Shot - Woi	36.1722	Spartan L	

Figura 4.7: Dataset final

De seguida, usámos a biblioteca MySQL Connector de Python de modo a podermos aceder à base de dados que será posteriormente criada e executar interrogações sobre ela. Para tal, eliminámos o esquema `gun_violence` caso ele existisse e criámos o mesmo através do ficheiro `create.sql`. Este ficheiro contém os comandos resultantes da ferramenta `Forward Engineering` fornecida pelo `MySQL Workbench`. O mesmo se pode comprovar pelo próximo excerto de código:

```

1 try:
2     cnx = mysql.connector.connect(user=config.user,
3                                   password=config.password,
4                                   host=config.host,
5                                   auth_plugin='mysql_native_password')
6
7     cursor = cnx.cursor()
8     cursor.execute("DROP SCHEMA IF EXISTS gun_violence;")
9     print("Dropped schema gun_violence if existed")
10
11     cnx._open_connection()
12
13     print("-----")
14
15     print("Creating gun_violence schema")
16     with open('create.sql', 'r') as f:
17         cursor.execute(f.read(), multi=True)
18         print("gun_violence created")
19
20 except mysql.connector.Error as err:
21     if err.errno == errorcode.ER_ACCESS_DENIED_ERROR:
22         print("Something is wrong with your user name or password")
23     else:
24         print(err)
25 else:
26     cnx.close()

```

De seguida, conectámo-nos novamente definindo a base de dados criada como a base de dados por omissão. Também reutilizámos um procedimento fornecido pelos docentes para povoar a nossa tabela `dim_date`.

```

1 cnx = mysql.connector.connect(user=config.user,
2                               password=config.password,
3                               host=config.host,
4                               database=config.database,
5                               auth_plugin='mysql_native_password')

```

```

6 cursor = cnx.cursor()
7 cursor.execute("DROP PROCEDURE IF EXISTS gun_violence.generate_Dates;")
8 queryProc = """
9     CREATE PROCEDURE gun_violence.generate_Dates(date_start DATE, date_end DATE)
10    BEGIN
11    WHILE date_start <= date_end DO
12        INSERT INTO gun_violence.dim_date (date, day, month, year) VALUES (date_start, day(
13        date_start), month(date_start), year(date_start));
14        SET date_start = date_add(date_start, INTERVAL 1 DAY);
15    END WHILE;
16    END;"""
17 cursor.execute(queryProc)

```

De forma a saber qual a data de início e fim para povoar a tabela `dim_date`, para cada linha do *dataset* verificámos qual a primeira data e para as restantes linhas fomos comparar se o valor da data atual era maior/menor do que a do histórico.

```

1 diff = str(datetime.datetime.today()-date)
2 diff_days = int(diff.split(' ')[0])
3 if meaningful_lines==0:
4     older_date=date
5     older_date_aux = diff_days
6     newer_date=date
7     newer_date_aux = diff_days
8 else:
9     if diff_days > older_date_aux:
10         older_date_aux = diff_days
11         older_date=date
12     if diff_days < newer_date_aux:
13         newer_date=date
14         newer_date_aux = diff_days

```

De forma a facilitar o povoamento do nosso *data warehouse*, resolvemos implementar uma abordagem semelhante ao método de povoamento realizado nas aulas práticas recorrendo à ferramenta **Table Data Import Wizard**. Tal como nas aulas, criámos uma tabela temporária auxiliar, denominada por *aux*, que irá conter todos os atributos necessários.

```

1 cursor.execute("""
2 CREATE TEMPORARY TABLE IF NOT EXISTS 'gun_violence'. 'aux' (
3     'id' INT NOT NULL AUTO_INCREMENT,
4     'incident_id' INT NULL,
5     'date' DATE NULL,
6     'state' TEXT NULL,
7     'city_or_county' TEXT NULL,
8     'address' TEXT NULL,
9     'n_killed' INT NULL,
10    'n_injured' INT NULL,
11    'gun_stolen' TEXT NULL,
12    'gun_type' TEXT NULL,
13    'incident_characteristics' TEXT NULL,
14    'latitude' DECIMAL(13,8) NULL,
15    'location_description' TEXT NULL,
16    'longitude' TEXT NULL,
17    'notes' TEXT NULL,
18    'participant_age' TEXT NULL,
19    'participant_age_group' TEXT NULL,
20    'participant_gender' TEXT NULL,
21    'participant_name' TEXT NULL,
22    'participant_relationship' TEXT NULL,
23    'participant_status' TEXT NULL,
24    'participant_type' TEXT NULL,
25    'state_house_district' INT NULL,
26    'state_senate_district' INT NULL,
27    PRIMARY KEY ('id'))
28 ENGINE = InnoDB;
29 """)

```


4.1 Extração

O processo de extração consiste em compreender, seleccionar e copiar os dados fonte para a Área de tratamento dos Dados.

Para tal, carregámos o *dataset* para podermos manipulá-lo, usufruindo da biblioteca *xlrd*.

```
1 print(f'Loading the dataset...')
2 book = xlrd.open_workbook('../dataset/gun-violence-data_01-2013_03-2018.xlsx')
3 sheet = book.sheet_by_index(0)
4
5 rows= sheet.nrows
```

4.2 Transformação

O processo de transformação de dados trata de efetuar uma limpeza dos dados, eliminação de campos inúteis, entre outros.

Assim, através da descrição dos atributos do *dataset* escolhido que se pode observar na Tabela 2.1, decidimos retirar os seguintes atributos: *incident_url*, *source_url*, *incident_url_fields_missing*, *congressional_district*, *sources* e *n_guns_involved*. O motivo desta remoção de atributos foi o facto de não serem relevantes para o projeto, à exceção do *n_guns_involved* que seria bastante útil caso estivesse de acordo com os dados apresentados e a maior parte dos valores não fossem nulos.

Para esta remoção de atributos e para o tratamento dos campos em branco, iterámos cada linha do *dataset* e guardámos os seus respetivos valores.

```
1 for row in range(1,rows):
2
3     if sheet.cell_type(row,0)==2 and sheet.cell_type(row,1)==3 and not(re.search("-[0-9].*",
4         str(sheet.cell_value(row,17)))) and not(re.search("District of Columbia",sheet.
5         cell_value(row,2),re.IGNORECASE)):
6         incident_id = int(sheet.cell_value(row,0))
7         date = datetime.datetime(*xlrd.xldate_as_tuple(sheet.cell_value(row,1), book.datemode
8         ))
9
10        #No ficheiro original encontra-se o c d i g o para obter a data mais antiga e mais
11        recente
12
13        state = sheet.cell_value(row,2)
14        city_or_county = sheet.cell_value(row,3)
15
16        address = sheet.cell_value(row,4).replace("'",'') if sheet.cell_type(row,4)==1 else "
17        N/A"
18        n_killed = int(sheet.cell_value(row,5)) if sheet.cell_value(row,5) != "" else -1
19        n_injured = int(sheet.cell_value(row,6)) if sheet.cell_value(row,6) != "" else -1
20        gun_stolen = sheet.cell_value(row,11)
21        gun_type = sheet.cell_value(row,12)
22
23        if sheet.cell_value(row,13)!="":
24            if sheet.cell_type(row,13)==1:
25                incident_characteristics = sheet.cell_value(row,13).replace("'",'')
26            elif sheet.cell_type(row,13)==3:
27                x = re.search("00:00:00",str(xlrd.xldate_as_datetime(sheet.cell_value(row,13),
28                    book.datemode)))
29                if x:
30                    incident_characteristics = (str(xlrd.xldate_as_datetime(sheet.cell_value(row
31                        ,13), book.datemode)).split(" ")[0])
32                else:
33                    incident_characteristics = (str(xlrd.xldate_as_datetime(sheet.cell_value(row
34                        ,13), book.datemode)).split(" ")[-1])
35            else:
36                incident_characteristics = sheet.cell_value(row,13)
37        else:
38            incident_characteristics = "N/A"
39
40        latitude = sheet.cell_value(row,14) if sheet.cell_value(row,14) != "" else -1
```

```

34     if sheet.cell_value(row,15)!="":
35         location_description = sheet.cell_value(row,15).replace(',', '') if sheet.cell_type(
row,15)==1 and sheet.cell_type(row,15)!=3 else sheet.cell_value(row,15)
36     else:
37         location_description = "N/A"
38     longitude = sheet.cell_value(row,16) if sheet.cell_value(row,16) != "" else -1
39
40     if sheet.cell_value(row,18)!="":
41         if sheet.cell_type(row,18)==1:
42             notes = sheet.cell_value(row,18).replace(',', '')
43         elif sheet.cell_type(row,18)==3:
44             x = re.search("00:00:00",str(xlrd.xldate_as_datetime(sheet.cell_value(row,18),
book.datemode)))
45             if x:
46                 notes = (str(xlrd.xldate_as_datetime(sheet.cell_value(row,18), book.datemode)).
split(" ")[0])
47             else:
48                 notes = (str(xlrd.xldate_as_datetime(sheet.cell_value(row,18), book.datemode)).
split(" ")[-1])
49             else:
50                 notes = sheet.cell_value(row,18)
51         else:
52             notes = "N/A"
53
54     participant_age_group = sheet.cell_value(row,20)
55     participant_gender = sheet.cell_value(row,21)
56     participant_name = sheet.cell_value(row,22).replace(',', '')
57     participant_relationship = sheet.cell_value(row,23)
58     participant_status = sheet.cell_value(row,24)
59     participant_type = sheet.cell_value(row,25)
60     state_house_district = int(sheet.cell_value(row,27)) if sheet.cell_type(row,27) == 2
else -1
61     state_senate_district = int(sheet.cell_value(row,28)) if sheet.cell_type(row,28) ==
2 else -1

```

Como podemos observar na linha 3 do excerto acima, para ser possível a inserção de uma linha, o primeiro campo (`incident_id`) tem de ser um número. Sendo que o tipo 3 da biblioteca `xlrd` não se refere apenas a números inteiros, usámos expressões regulares para determinar se o mesmo não era um valor do tipo `float`. Sendo a componente temporal extremamente importante nos *data warehouses*, a segunda célula tem de ser uma data. Por fim, reparámos que o atributo `state` por vezes tinha valor District of Columbia. Ora, este não é um dos 50 estados dos EUA, mas sim um distrito federal. Assim sendo e tendo em conta que apenas existiam 2938 registos com esse valor (representam 1% do *dataset*), decidimos remover essas linhas em vez de colocar o valor desse campo igual a N/A.

Para valores cujos campos no Excel se encontravam em branco, decidimos que teriam o valor de N/A caso o tipo desse atributo fosse `varchar` e -1 se este fosse do tipo `integer`.

Para atributos que eram `varchar`, mas que apresentavam uma data ou hora tivemos de passar o valor para o tipo `datetime` da biblioteca `xlrd` e, posto isto, partir conforme o valor fosse uma data ou uma hora (linhas 44 a 48 do excerto acima).

De forma a guardar os tipos das armas presentes no *dataset*, adicionámos o tipo ao conjunto `gun_type_set`, partindo previamente conforme um tipo de separador.

```

1 gun_type_set = set()
2
3     if gun_type != "":
4         if '|' in gun_type:
5             splitted = gun_type.split("|")
6             for gun in splitted:
7                 type = gun.split("::")[-1]
8                 gun_type_set.add(type)
9         else:
10            splitted = gun_type.split("|")
11            for gun in splitted:
12                type = gun.split("::")[-1]
13                gun_type_set.add(type)

```

No atributo `participant_age` fizemos um tratamento especial, pois, como os separadores eram maioritariamente `::`, o valor passava para uma hora, o que é errado. Assim, tratámos de repor o valor original.

```

1 if sheet.cell_type(row,19) != 2:
2     if sheet.cell_type(row,19) == 3:
3         x = sheet.cell_value(row,19) # a float
4         x = int(x * 24 * 3600) # convert to number of seconds
5         participant_age = f'{x//3600}:{(x%3600)//60}'
6     else:
7         participant_age = sheet.cell_value(row,19)

```

O tratamento dos participantes e armas será feito com o carregamento para as dimensões `dim_participant` e `dim_gun`, respetivamente. O mesmo será explicado na secção seguinte.

Por fim, inserimos os dados na tabela `aux`, mencionada na secção 4.

```

1 cursor.execute(f' INSERT INTO gun_violence.aux (\n'
2 f'incident_id, date, state, city_or_county, address, n_killed,\n'
3 f'n_injured, gun_stolen, gun_type, incident_characteristics,\n'
4 f'latitude, location_description, longitude,\n'
5 f'notes, participant_age,\n'
6 f'participant_age_group, participant_gender,\n'
7 f'participant_name, participant_relationship,\n'
8 f'participant_status, participant_type,\n'
9 f'state_house_district,state_senate_district) VALUES\n'
10
11 f'("{incident_id}", "{date}", "{state}", "{city_or_county}", "{address}", {n_killed}
12 },\n'
13 f'{n_injured}, "{gun_stolen}", "{gun_type}", "{incident_characteristics}",\n'
14 f'"{latitude}", "{location_description}", {longitude},\n'
15 f'"{notes}", "{participant_age}",\n'
16 f'"{participant_age_group}", "{participant_gender}",\n'
17 f'"{participant_name}", "{participant_relationship}",\n'
18 f'"{participant_status}", "{participant_type}",\n'
19 f'"{state_house_district}", {state_senate_district});'

```

4.3 Carregamento

Este processo, tal como o nome indica, trata de carregar os dados já corretos e claros para o *data warehouse*.

Assim, para povoar a tabela `dim_date`, invocámos apenas o procedimento mencionado na secção 4 com as datas referentes ao *dataset*.

```

1 print("Populating dim_date...")
2 cursor.execute(f'CALL gun_violence.generate_Dates("{older_date}", "{newer_date}");')
3 print("done")

```

Sendo que a faixa etária de um participante poderia ser Adult 18+, Child 0-11, Teen 12-17 ou N/A, decidimos inseri-los manualmente.

```

1 print("Populating dim_participant_age_group...")
2 cursor.execute("insert into dim_participant_age_group (dim_participant_age_group_id,
3 class_age_group) VALUES (1,'Adult 18+');")
4 cursor.execute("insert into dim_participant_age_group (dim_participant_age_group_id,
5 class_age_group) VALUES (2,'Child 0-11');")
6 cursor.execute("insert into dim_participant_age_group (dim_participant_age_group_id,
7 class_age_group) VALUES (3,'Teen 12-17');")
8 cursor.execute("insert into dim_participant_age_group (dim_participant_age_group_id,
9 class_age_group) VALUES (4,'N/A');")
10 print("done")

```

Analogamente, realizámos o povoamento da tabela `dim_gun_stolen`, visto que os valores apenas podem ser Unknown, Stolen ou Not-stolen.

```

1 print("Populating dim_gun_stolen...")
2 cursor.execute("insert into dim_gun_stolen (dim_gun_stolen_id,class_stolen) VALUES (1,'
3 Unknown');")
4 cursor.execute("insert into dim_gun_stolen (dim_gun_stolen_id,class_stolen) VALUES (2,'
5 Stolen');")
6 cursor.execute("insert into dim_gun_stolen (dim_gun_stolen_id,class_stolen) VALUES (3,'
7 Not-stolen');")

```

Para povoar a tabela `dim_gun_type` percorremos então o conjunto criado para tal e inserimos os valores na mesma.

```

1 print("Populating dim_gun_type...")
2 x=1
3 for val in gun_type_set:
4     cursor.execute(f'INSERT INTO gun_violence.dim_gun_type (dim_gun_type_id,class_type)
5         VALUES ({x},"{val}");\n')
6     x+=1
7 print("done")

```

Fazendo inserções a partir de projeções da tabela aux, povoamos as tabelas de dimensão dim_state_district, dim_incident_info, dim_location e a tabela de factos facts_gun_incident. Exemplo:

```

1 cursor.execute("""
2     INSERT INTO facts_gun_incident (incident_id, n_killed, n_injured, dim_date_id,
3     dim_incident_info_id, dim_location_id)
4     SELECT incident_id, n_killed, n_injured, t1.dim_date_id, incident_id, incident_id
5     FROM gun_violence.aux t
6     LEFT JOIN dim_date t1
7     ON t.date=t1.date
8 """)

```

```

1 cursor.execute(f'select gun_stolen, gun_type, incident_id from gun_violence.aux where id
2     ={idAux}')
3 gun = cursor.fetchone()
4 gun_stolen, gun_type, incident_id = gun

```

Para povoar a tabela dim_gun foi necessário obter os campos gun_stolen, gun_type e incident_id da tabela aux. Os campos gun_type e gun_stolen foram divididos pelo respetivo separador de campo e conjugados entre si, pois estes campos possuem mais do que uma entidade (exemplo de uma linha da tabela gun_type: 1::Deagle || 2::AK-47). O código seguinte tratou desse processo:

```

1 separator = "#"
2 if gun_stolen != "" and gun_type != "":
3     if '||' in gun_type:
4         separator = "||"
5         scn_separator = "::"
6
7     elif '|' in gun_type:
8         separator = "|"
9         scn_separator = ":"
10
11     else:
12         if '::' in gun_type:
13             scn_separator = "::"
14         else:
15             scn_separator = ":"
16
17 splitted_type = gun_type.split(separator)
18 splitted_stolen = gun_stolen.split(separator)
19
20 for i in range(len(splitted_type)):
21     stolen_gun = splitted_stolen[i].split(scn_separator)[-1]
22     type_gun = splitted_type[i].split(scn_separator)[-1]
23
24     insert_dim_gun(cursor, stolen_gun, type_gun)

```

Depois de divididas as separações de campo, foi então adicionada uma nova linha à tabela dim_gun.

Para a tabela dim_participant foi usado o mesmo mecanismo de separação. A única diferença foram os campos da tabela aux selecionados e o facto de existirem vários participantes envolvidos com diversas características, logo foi necessário descobrir quantos participantes existiam e depois caracterizar esses participantes pelo seu respetivo identificador, como podemos ver no código seguinte:

```

1
2
3 cursor.execute(f'select participant_gender, participant_name, participant_relationship,
4     participant_status, participant_type, participant_age, participant_age_group from aux
5     where id={idAux};')
6 participants = cursor.fetchall()
7
8 aux = tuple(filter(lambda x: len(x)>1,participants))
9
10 selected = aux[0] if len(aux) > 0 else ""

```

```

10 separator = ""
11
12 if '||' in selected:
13     separator = "||"
14     scn_separator = "::"
15
16 elif '|' in selected:
17     separator = "|"
18     scn_separator = ":"
19
20 else:
21     if '::' in selected:
22         scn_separator = "::"
23     else:
24         scn_separator = ":"
25
26 if separator != "":
27     genders, names, relationships, statuss, types, ages, age_groups = tuple(map(lambda x:
28         to_dict(x, separator, scn_separator), participants))
29     max_len = reduce(lambda e1, e2: max(e1, len(e2.split(separator))), participants, 0)
30
31     for i in range(max_len):
32         cursor.execute(f'select incident_id from aux where id={idAux};')
33         incident_id, = cursor.fetchone()
34
35         gender = genders[str(i)] if str(i) in genders else "N/A"
36         name = names[str(i)] if str(i) in names else "N/A"
37         relationship = relationships[str(i)] if str(i) in relationships else "N/A"
38         status = statuss[str(i)] if str(i) in statuss else "N/A"
39         ptype = types[str(i)] if str(i) in types else "N/A"
40         age = ages[str(i)] if str(i) in ages else -1
41         age_group = age_groups[str(i)] if str(i) in age_groups else "N/A"
42
43         # para buscar id age group
44         if age_group != "N/A":
45             cursor.execute(f'select dim_participant_age_group_id from
46                 dim_participant_age_group where class_age_group="{age_group}";')
47             id_age_group, = cursor.fetchone()
48         else:
49             id_age_group = 4
50
51         cursor.execute(f'INSERT INTO gun_violence.dim_participant (gender,name,relationship
52             ,status,type,dim_participant_age_group_id,age,facts_gun_incident_id) VALUES ("{gender}
53            ",{name},"{relationship},"{status},"{ptype},{id_age_group},{age},{incident_id})
54             ;')
55
56     idAux+=1
57     print("done")
58     cursor.close()
59     cnx.commit()

```

5 Indicadores de *Business Intelligence*

Por fim, iremos apresentar onze exemplos de indicadores de *Business Intelligence* relevantes criados com o auxílio da ferramenta Tableau. Optámos por utilizar este *software*, uma vez que consegue lidar com um grande volume de dados com melhor desempenho quando comparado com a ferramenta PowerBI e também devido ao facto de estarmos mais familiarizados com o mesmo.

Na figura 5.1, podemos observar o primeiro indicador que, por sua vez, apresenta o número total de incidentes por estado. Daqui, podemos retirar que Califórnia e Illinois são alguns dos estados com mais incidentes e, em contrapartida, Hawaii e Vermont são alguns dos estados com menos incidentes.

Número total de incidentes por estado

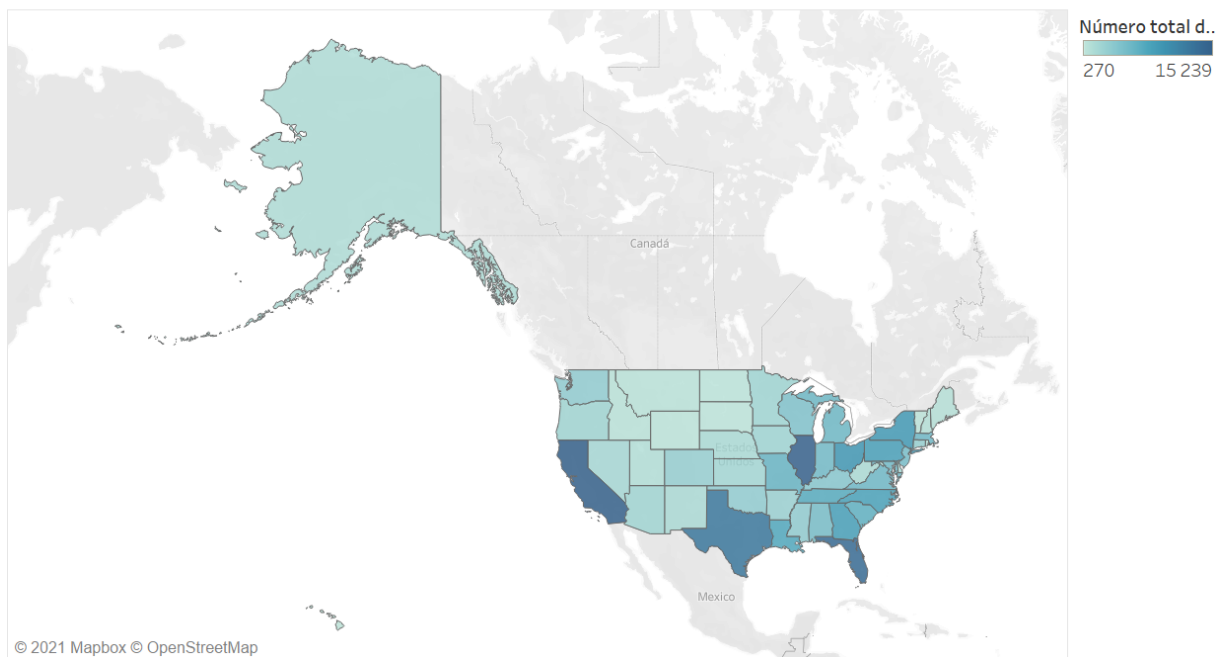


Figura 5.1: Número total de incidentes por estado

Na figura 5.2, podemos observar o segundo indicador que, por sua vez, apresenta o número de suspeitos por faixa etária. Daqui, podemos retirar que a maior parte dos suspeitos já é adulto. Todavia, existe uma porção significativa de adolescentes com o estatuto de suspeito.

Número de suspeitos por faixa etária

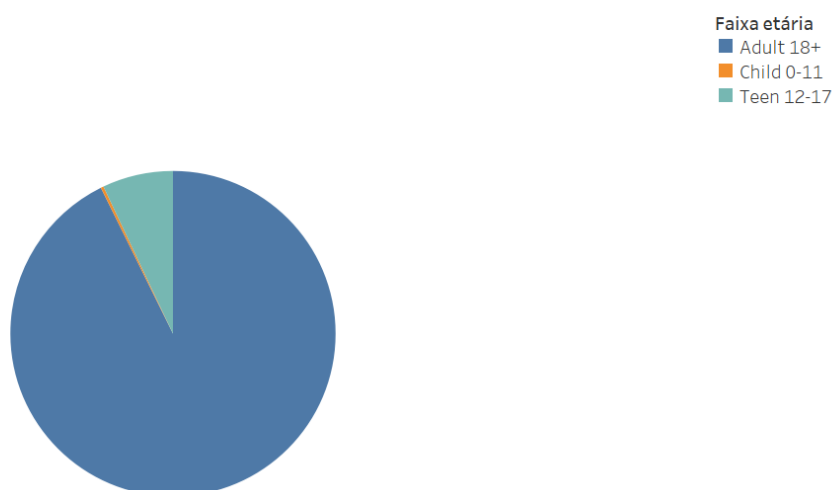


Figura 5.2: Número de suspeitos por faixa etária

Na figura 5.3, podemos observar o terceiro indicador que, por sua vez, apresenta o número de presos por estado. Nesta imagem, é possível observar uma clara maioria no estado da Pennsylvania com 16 presos.

Número de presos por estado

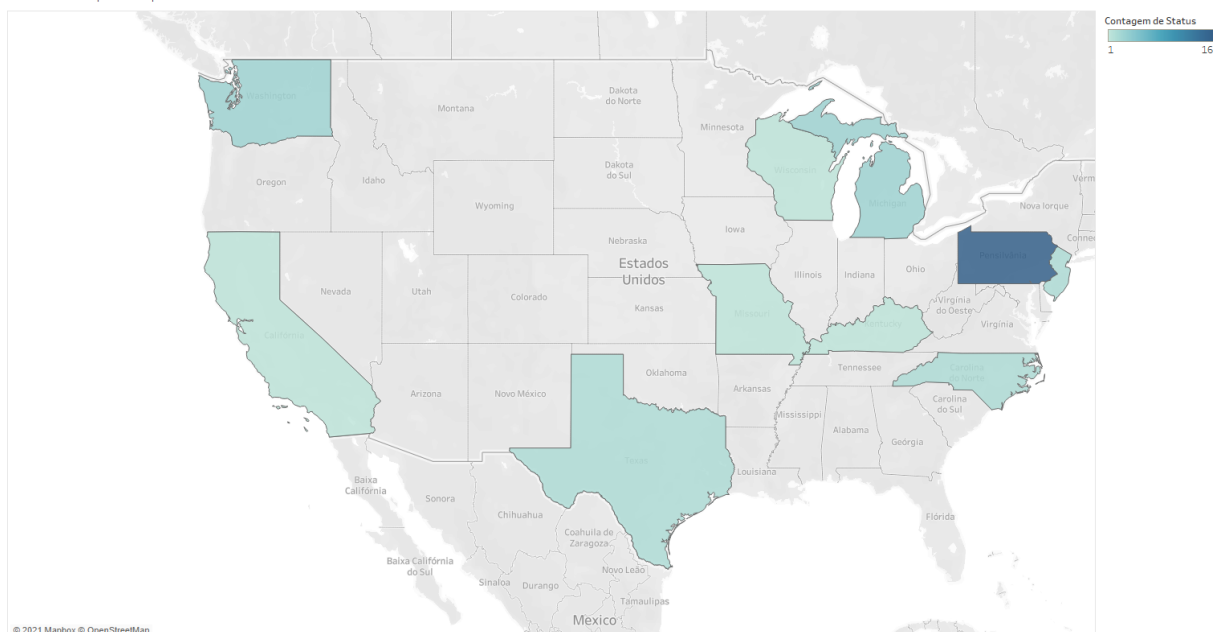


Figura 5.3: Número de presos por estado

Na figura 5.4, podemos observar o quarto indicador que, por sua vez, apresenta o número total de incidentes por ano e mês. Daqui, podemos retirar que as situações em que houve um maior número de incidentes ocorreram nos meses mais quentes (julho e agosto de 2014).

Número total de incidentes por ano e mês

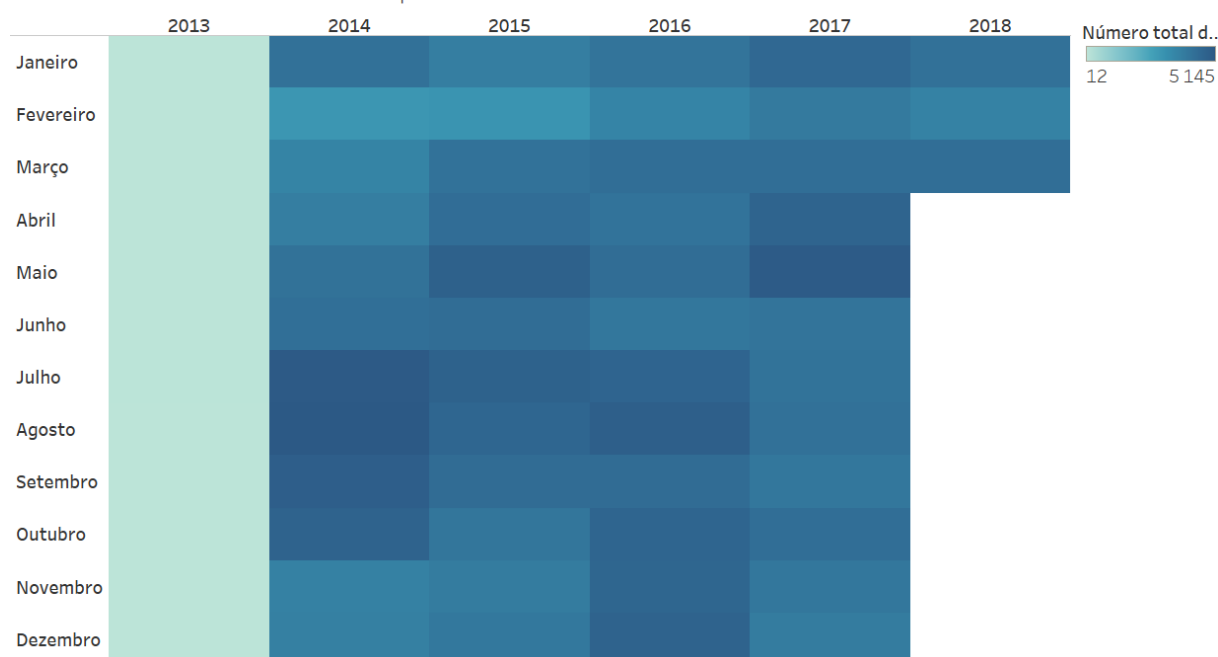


Figura 5.4: Número total de incidentes por ano e mês

Na figura 5.5, podemos observar o quinto indicador que, por sua vez, apresenta o número de incidentes com cada tipo de arma. Daqui, podemos retirar que na maior parte dos incidentes não se sabe que tipo de arma foi usada e, em contrapartida, 28 gauge e 300 Win são alguns exemplos dos tipos de armas menos usadas.

Número de incidentes com cada tipo de arma



Figura 5.5: Número de incidentes com cada tipo de arma

Na figura 5.6, podemos observar o sexto indicador que, por sua vez, apresenta o número de suspeitos por relação e género. Nesta imagem, é possível constatar que o maior número de suspeitos do sexo masculino teve origem em acontecimentos de assalto à mão armada (**Armed Robbery**), já no sexo feminino o maior

número de suspeitas teve origem em acontecimentos relacionados com família (Family). É também possível concluir que a grande maioria dos suspeitos são do sexo masculino.

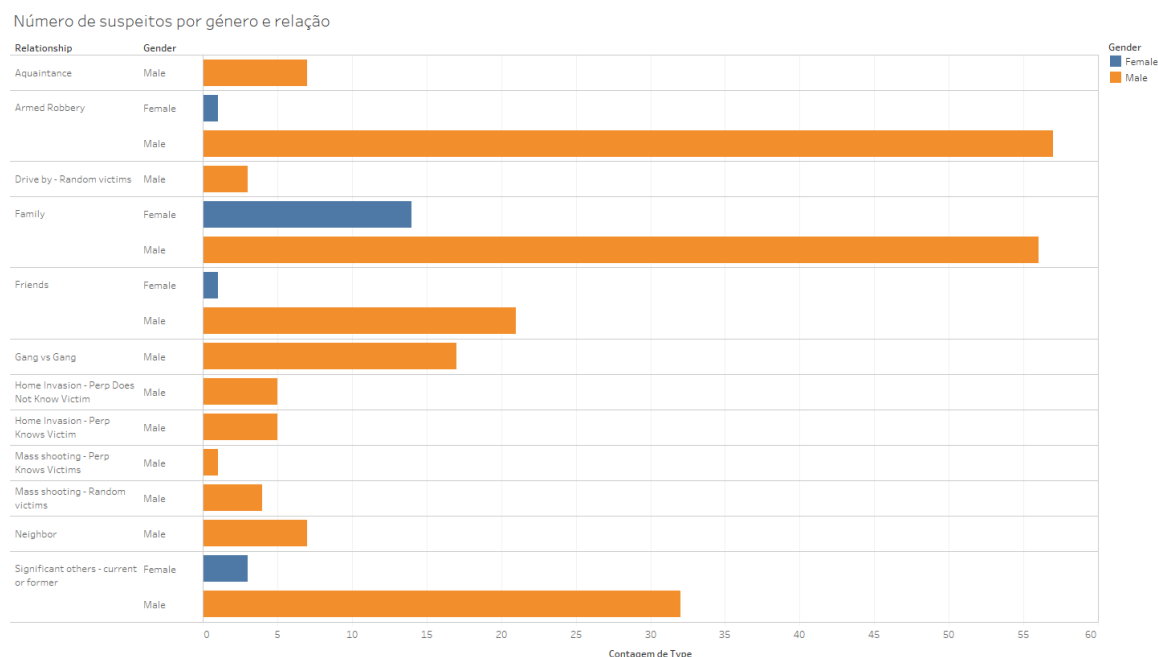


Figura 5.6: Número de suspeitos por relação e género

Na figura 5.7, podemos observar o sétimo indicador que, por sua vez, apresenta a média de mortes por incidente por estado. Daqui, podemos retirar que Arizona é o estado onde, em média, houve mais mortes por incidente e, em contrapartida, Rhode Island é aquele onde houve menos. Contudo, podemos verificar que a média de todos eles está abaixo de 1 e, portanto, podemos concluir que foram poucos os incidentes onde, efetivamente, morreram pessoas.

Média de mortes por incidente por estado

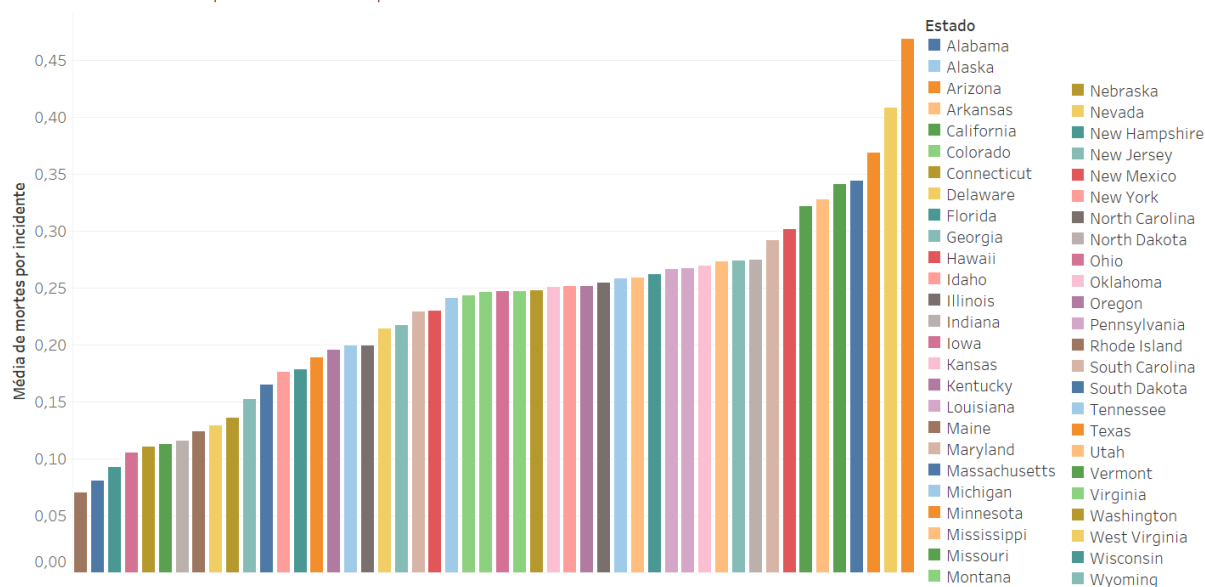


Figura 5.7: Média de mortes por incidente por estado

Na figura 5.8, podemos observar o oitavo indicador que, por sua vez, apresenta o número de mortos e o número de feridos por género. Podemos observar que, em termos de feridos (Injured), há uma enorme maioria do sexo masculino em relação ao sexo feminino. O mesmo acontece em relação ao número de

mortos. De notar que o número de homens e mulheres ferido(a)s é superior em relação ao número de mortos.

Número de mortos vs Número de feridos por género

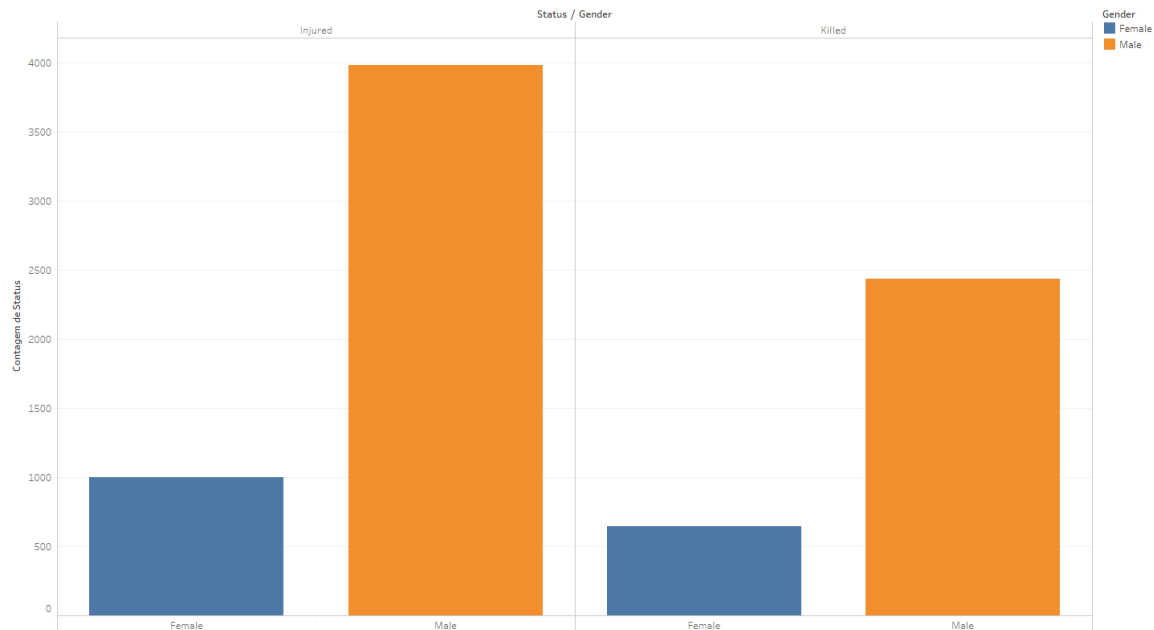


Figura 5.8: Número de mortos e feridos por género

Na figura 5.9, podemos observar o nono indicador que, por sua vez, apresenta o número de suspeitos por estado após o acontecimento. Podemos observar que 31 suspeitos foram presos, 275 foram feridos e 378 foram mortos. Daqui, é possível concluir que a grande maioria dos suspeitos acabou por ser morta ou ferida em vez de presa.

Número de suspeitos presos vs feridos vs mortos



Figura 5.9: Número de suspeitos presos, feridos e mortos

Na figura 5.10, podemos observar o décimo indicador que, por sua vez, apresenta o número de vítimas dos acontecimentos por género. Há uma maioria de vítimas do sexo masculino (7854) em relação ao número de vítimas do sexo feminino (2782).

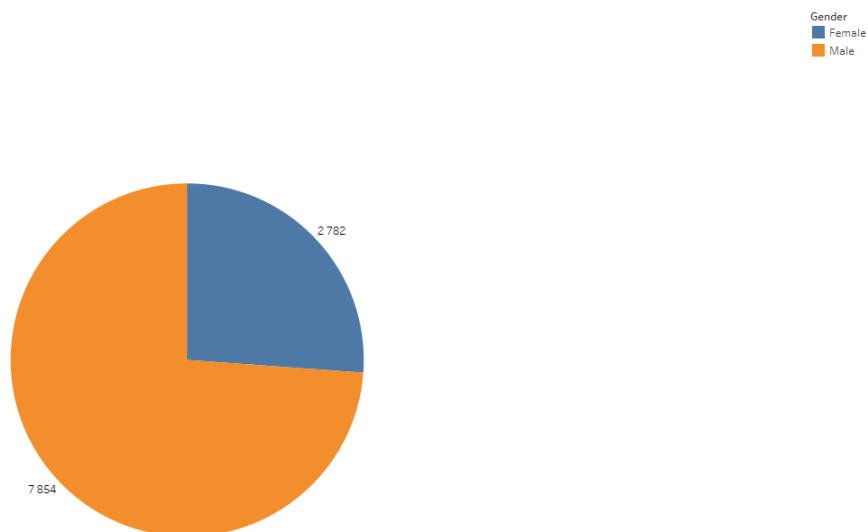


Figura 5.10: Número de vítimas por género

Na figura 5.11, podemos observar o décimo primeiro e último indicador que, por sua vez, apresenta a média de idades de suspeitos que roubaram e não roubaram armas. É possível observar que a média de idades dos suspeitos é maior nos casos de arma roubada (21.714) em relação aos casos de arma não roubada (19).

Média de idades dos
suspeitos que
roubaram/não
roubaram armas

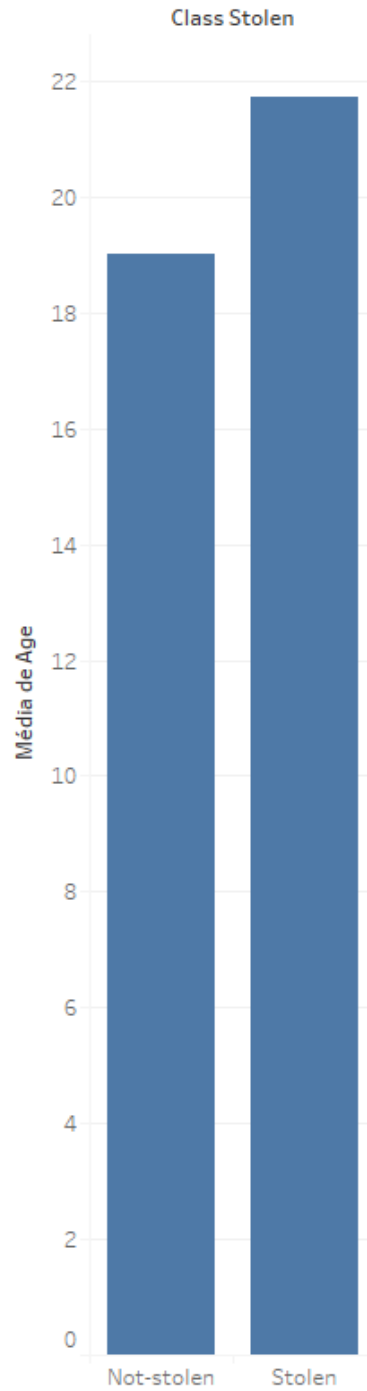


Figura 5.11: Média de idades de suspeitos que roubaram e não roubaram armas

6 Conclusão

Após um estudo dos dados do *dataset* escolhido relativo à área da criminalidade, foi desenhado um *data warehouse*, com base no método dos 4 passos de Kimball, que permitisse acolher os dados da melhor forma. Posteriormente, foram aplicadas edições e ajustes aos dados com processos de *ETL* de modo a que estes pudessem ser inseridos de forma consistente no *data warehouse*. Feito isto, foram criados os indicadores de *Business Intelligence* utilizando a ferramenta de suporte à decisão Tableau que, por sua vez, permitiu obter uma visão geral dos acontecimentos em termos históricos.

Durante a execução deste trabalho, foram várias as dificuldades enfrentadas. Quanto à escolha do *dataset*, o principal desafio foi encontrar um que satisfizesse uma série de requisitos (como a presença de atributos com datas ou a presença de atributos que permitissem fazer bons indicadores de *Business Intelligence*) para que este pudesse ser utilizado num *data warehouse*. Já no processo de *ETL*, as principais dificuldades surgiram no carregamento do *dataset* em Python. Visto que o *dataset* original se encontrava no formato *CSV*, qualquer mudança realizada sobre ele não era persistente. Assim, tivemos de fazer a mudança de código para permitir a manipulação de dados de ficheiros em Excel. Também tivemos dificuldade em separar o valor de cada participante e arma para popular as dimensões `dim_participant` e `dim_gun`, respetivamente, mas rapidamente foram ultrapassadas através de funções de ordem superior, como `filter` e `reduce`. Por último, relativamente ao processo de criação de indicadores de *Business Intelligence*, as principais adversidades consistiram na decisão de quais implementar, uma vez que estes teriam de apresentar informação relevante para um processo de decisão e análise.