Data processing for big data with Apache Spark

Thanasak Harisombut

Date: 05/09/2020

Version: 1.0

Environment: Python 3.7.4, Apache Spark, and Jupyter notebook

1. Working with RDD

1.1 Data preparation and loading

1.1.1

In [1]:
```
# Import SparkConf class into program
from pyspark import SparkConf

# local[*]: run Spark in local mode with as many working processors
as logical cores on your machine
master = "local[*]"

# The `appName` field is a name to be shown on the Spark cluster UI
page
app_name = "Project 1"

# Setup configuration parameters for Spark
spark_conf = SparkConf().setMaster(master).setAppName(app_name)

# Import SparkContext and SparkSession classes
from pyspark import SparkContext # Spark
from pyspark.sql import SparkSession # Spark SQL

# Using SparkSession
spark = SparkSession.builder.config(conf=spark_conf).getOrCreate()
sc = spark.sparkContext
sc.setLogLevel('ERROR')
```

1.1.2

In [2]:
```
unit_rdd = sc.textFile('data/201[5-9]_DATA_SA_Units.csv')
```

1.1.3

In [3]:
```
crash_rdd = sc.textFile('data/201[5-9]_DATA_SA_Crash.csv')
```

1.1.4

In [4]:
```
unit_header = unit_rdd.first()
unit_rdd = unit_rdd.filter(lambda row: row != unit_header)   #filte
r out header

print('Total count of unit:', unit_rdd.count())
print('First 10 records: ')
unit_rdd.take(10)
```

```
Total count of unit: 153854
First 10 records:
```

Out[4]:
```
['"2016-1-15/08/2019","01",0,"SA","OMNIBUS","2011","North","Male",
"056","SA","HR","Full","Not Towing","Straight Ahead","010","5121",
,',
 '"2016-1-15/08/2019","02",1,,"Pedestrian on Road",,"East","Male",
"072",,,,,"Walking on Road",,"5084",,',
 '"2016-2-15/08/2019","01",0,"SA","Motor Cars - Sedan","2004","Unk
nown","Female","023","SA","C ","Full","Not Towing","Straight Ahead
","001","5087",,',
 '"2016-2-15/08/2019","02",0,"SA","Station Wagon","2008","Unknown"
,"Male","040","SA","C ","Full","Not Towing","Straight Ahead","001"
,"5084",,',
 '"2016-3-15/08/2019","01",0,"SA","RIGID TRUCK LGE GE 4.5T","1990"
,"South","Unknown","XXX","SA","MR","Provisional 2","Not Towing","S
traight Ahead","001","5115",,',
 '"2016-3-15/08/2019","02",0,"SA","Panel Van","2013","South","Male
","023","SA","C ","Full","Not Towing","Straight Ahead","001","5110
",,',
 '"2016-4-15/08/2019","01",0,"SA","Station Wagon","2002","East","F
emale","033","SA","C ","Full","Not Towing","Straight Ahead","001",
"5169",,',
 '"2016-4-15/08/2019","02",0,"UNKNOWN","Other Defined Special Vehi
cle","XXXX","North","Unknown","XXX","UNKNOWN","XX","Unknown","Unkn
own","Reversing","001","XXXX",,',
 '"2016-5-15/08/2019","01",1,"SA","Motor Cars - Sedan","1997","Sou
th East","Male","042","SA","C ","Full","Not Towing","Right Turn","
001","XXXX",,',
 '"2016-5-15/08/2019","02",2,"SA","Utility","2015","North East","M
ale","059","SA","MC","Full","Not Towing","Straight Ahead","002","5
114",,']
```

In [5]:
```
crash_header = crash_rdd.first()
crash_rdd = crash_rdd.filter(lambda row: row != crash_header)   #fi
lter out header

print('Total count of unit:', crash_rdd.count())
print('First 10 records: ')
crash_rdd.take(10)
```

```
Total count of unit: 72006
First 10 records:
```

Out[5]:

```
['"2019-1-8/07/2020","2 Metropolitan","HAMPSTEAD GARDENS","5086","
CITY OF PORT ADELAIDE ENFIELD",2,0,0,0,0,2019,"June","Wednesday","
11:15 am","060","Cross Road","Straight road","Level","Not Applicab
le","Sealed","Dry","Not Raining","Daylight","Right Angle","01","Dr
iver Rider","1: PDO","Give Way Sign","","",1331810.03,1676603.26,"
13318101676603"',
 '"2019-2-8/07/2020","2 Metropolitan","DRY CREEK","5094","CITY OF
SALISBURY",2,0,0,0,0,2019,"January","Tuesday","12:49 am","090","Di
vided Road","Straight road","Level","Not Applicable","Sealed","Dry
","Not Raining","Night","Rear End","02","Driver Rider","1: PDO","N
o Control","","",1328376.2,1682942.63,"13283761682943"',
 '"2019-3-8/07/2020","2 Metropolitan","MILE END","5031","CITY OF W
EST TORRENS",2,1,0,0,1,2019,"January","Tuesday","12:00 am","060","
Divided Road","Straight road","Level","Not Applicable","Sealed","D
ry","Not Raining","Night","Hit Pedestrian","01","Driver Rider","2:
MI","No Control","","",1325819.68,1670994.26,"13258201670994"',
 '"2019-4-8/07/2020","2 Metropolitan","PARALOWIE","5108","CITY OF
SALISBURY",2,1,0,1,0,2019,"January","Tuesday","12:05 am","050","No
t Divided","CURVED, VIEW OPEN","Level","Not Applicable","Sealed","
Dry","Not Raining","Night","Hit Fixed Object","01","Driver Rider",
"3: SI","No Control","","",1328320.6,1690237.08,"13283211690237"',
 '"2019-5-8/07/2020","2 Metropolitan","MOUNT BARKER","5251","DC MT
.BARKER.                ",2,0,0,0,0,2019,"January","Tuesday","05:
15 am","110","Divided Road","Straight road","Slope","Not Applicabl
e","Sealed","Dry","Not Raining","Night","Hit Animal","02","Animal"
,"1: PDO","No Control","","",1353279.99,1655645.15,"13532801655645
"',
 '"2019-6-8/07/2020","2 Metropolitan","TORRENSVILLE","5031","CITY
OF WEST TORRENS",2,1,0,0,1,2019,"January","Tuesday","07:00 am","05
0","Divided Road","Straight road","Level","Not Applicable","Sealed
","Dry","Not Raining","Daylight","Hit Fixed Object","01","Driver R
ider","2: MI","No Control","","",1324652.75,1672027.64,"1324653167
2028"',
 '"2019-7-8/07/2020","2 Metropolitan","BEDFORD PARK","5042","CC MI
TCHAM.                  ",2,3,0,0,3,2019,"January","Tuesday","09:
40 am","050","Cross Road","Straight road","Level","Not Applicable"
,"Sealed","Dry","Not Raining","Daylight","Right Angle","02","Drive
r Rider","2: MI","Traffic Signals","","",1325156.2,1660414.38,"132
51561660414"',
 '"2019-8-8/07/2020","3 Country","WYE","5291","DISTRICT COUNCIL OF
GRANT",1,1,0,0,1,2019,"January","Tuesday","12:15 pm","110","Not Di
vided","CURVED, VIEW OPEN","Level","Not Applicable","Sealed","Dry"
,"Not Raining","Daylight","Roll Over","01","Driver Rider","2: MI",
"No Control","","",1517347.8,1321979.33,"15173481321979"',
 '"2019-9-8/07/2020","3 Country","MOUNT GAMBIER","5290","CC MT.GAM
BIER.                ",1,1,0,0,1,2019,"January","Tuesday","11:45 a
m","050","T-Junction","Straight road","Bottom of Hill","Not Applic
able","Sealed","Dry","Not Raining","Daylight","Roll Over","01","Dr
iver Rider","2: MI","Stop Sign","","",1510220.05,1340472.35,"15102
201340472"',
 '"2019-10-8/07/2020","3 Country","OVERLAND CORNER","5330","THE BE
RRI BARMERA COUNCIL",1,1,0,1,0,2019,"January","Tuesday","12:30 pm"
,"080","Not Divided","CURVED, VIEW OPEN","Slope","Not Applicable",
"Sealed","Dry","Not Raining","Daylight","Roll Over","01","Driver R
ider","3: SI","No Control","","",1492599.9,1749395.71,"14926001749
396"']
```

1.2 Data Partitioning in RDD

1.2.1

How many partitions do the above RDDs have?

In [6]:
```
print("Unit RDD Partition number:", unit_rdd.getNumPartitions())
print("Crash RDD Partition number:", crash_rdd.getNumPartitions())
```

```
Unit RDD Partition number: 5
Crash RDD Partition number: 5
```

How is the data in these RDDs partitioned by default, when we do not explicitly specify any partitioning strategy?

The Spark will consider the file size, block size, and calculate the number of the partition. Furthermore, the data will be distributed randomly equal to all partitions.

In [7]:
```
def print_partition(rdd):
    numPartitions = rdd.getNumPartitions()
    partitions = rdd.glom().collect()
    print(f"NUMBER OF PARTITIONS: {numPartitions}")
    for index, partition in enumerate(partitions):
        # show partition if it is not empty
        if len(partition) > 0:
            print(f"- Partition {index}: {len(partition)} records")
print('####### Unit RDD #######')
print_partition(unit_rdd)
print('\n####### Crash RDD #######')
print_partition(crash_rdd)
```

```
####### Unit RDD #######
NUMBER OF PARTITIONS: 5
- Partition 0: 35861 records
- Partition 1: 28163 records
- Partition 2: 33084 records
- Partition 3: 27713 records
- Partition 4: 29033 records

####### Crash RDD #######
NUMBER OF PARTITIONS: 5
- Partition 0: 12964 records
- Partition 1: 16775 records
- Partition 2: 13237 records
- Partition 3: 13599 records
- Partition 4: 15431 records
```

1.2.2

In [8]:
```
# a) Create a key Value pair with Lic Sate
trnf_unit_rdd = unit_rdd.map(lambda x: x.split(",")) \
    .map(lambda x: [s.strip("\"") for s in x]).map(lambda x: (x[9],
x[0:9] + x[10:]))
```

In [9]:
```
# b) Hash Function and implement Hash Partitioning
def hash_function(key):
    if key.lower() == 'sa':
        return 0
    else:
        return 1
no_of_partitions = 2
partitioned_unit_rdd = trnf_unit_rdd.partitionBy(no_of_partitions,
hash_function)
```

In [10]:
```
# c) checking unit's partition by hash function
print('####### Hash Partitioned Unit RDD #######')
print_partition(partitioned_unit_rdd)
```

```
####### Hash Partitioned Unit RDD #######
NUMBER OF PARTITIONS: 2
- Partition 0: 109684 records
- Partition 1: 44170 records
```

Answer

For the data skewness, we can see that the first partition (SA State) has a significantly larger amount than the second partition (Other States).

Thus, it will have a poorly performance once working on this structure.

1.3 Query/Analysis

1.3.1

In [11]:
```
filtered_driver_rdd = unit_rdd.map(lambda x: x.split(",")) \
                      .map(lambda x: (x[7].strip("\""), x[8].strip("\
""))) \
                      .filter(lambda x: x[1].isnumeric() and (x[0].lo
wer()=="male" or x[0].lower()=="female")) \
                      .map(lambda x: (x[0], int(x[1])))

avg_age_rdd = filtered_driver_rdd.mapValues(lambda v: (v, 1)) \
                                 .reduceByKey(lambda a,b: (a[0]+b[0]
, a[1]+b[1])) \
                                 .mapValues(lambda v: v[0]/v[1])

print( "The average male age: ", round(avg_age_rdd.collectAsMap()['
Male'],2))
print( "The average female age: ", round(avg_age_rdd.collectAsMap()
['Female'],2))
```

```
The average male age:  40.98
The average female age:  40.39
```

1.3.2

In [12]:
```
veh_involved_rdd = unit_rdd.map(lambda x: x.split(",")) \
                   .map(lambda x: (x[3].strip("\""),x[5].strip("\""),x
[4].strip("\""))) \
                   .filter(lambda x: x[1].isnumeric()) \
                   .sortBy(lambda x: x[1], ascending=True)
the_oldest_v = veh_involved_rdd.first()
the_newest_v = veh_involved_rdd.collect()[veh_involved_rdd.count()-
1]

veh_involved_rdd = veh_involved_rdd.filter(lambda x: x==the_oldest_
v or x ==the_newest_v).distinct() \
                                   .sortBy(lambda x: x[1], ascendi
ng=True)

print("The oldest vehicle year involved in accident:", veh_involved
_rdd.first())
print("The newest vehicle year involved in accident:", veh_involved
_rdd.collect()[veh_involved_rdd.count()-1])
```

```
The oldest vehicle year involved in accident: ('VIC', '1900', 'Mot
or Cycle')
The newest vehicle year involved in accident: ('SA', '2019', 'Moto
r Cars - Sedan')
```

2. Working with DataFrame

2.1 Data Preparation and Loading

2.1.1 Load all data into dataframes

In [13]:
```
# Read csv file and load into a dataframe
unit_file_name = ['data/20'+ str(x) +'_DATA_SA_Units.csv' for x in
range(15,20)]
crash_file_name = ['data/20'+ str(x) +'_DATA_SA_Crash.csv' for x in
range(15,20)]
unit_df = spark.read.csv(unit_file_name, header=True)
crash_df = spark.read.csv(crash_file_name, header=True)
```

2.1.2 Display the schema

In [14]:
```
print(f"####### UNITS INFO:")
unit_df.printSchema()
print(f"####### CRASH INFO:")
crash_df.printSchema()
```

```
####### UNITS INFO:
root
 |-- REPORT_ID: string (nullable = true)
 |-- Unit No: string (nullable = true)
 |-- No Of Cas: string (nullable = true)
 |-- Veh Reg State: string (nullable = true)
 |-- Unit Type: string (nullable = true)
 |-- Veh Year: string (nullable = true)
 |-- Direction Of Travel: string (nullable = true)
 |-- Sex: string (nullable = true)
 |-- Age: string (nullable = true)
 |-- Lic State: string (nullable = true)
 |-- Licence Class: string (nullable = true)
 |-- Licence Type: string (nullable = true)
 |-- Towing: string (nullable = true)
 |-- Unit Movement: string (nullable = true)
 |-- Number Occupants: string (nullable = true)
 |-- Postcode: string (nullable = true)
 |-- Rollover: string (nullable = true)
 |-- Fire: string (nullable = true)

####### CRASH INFO:
root
 |-- REPORT_ID: string (nullable = true)
 |-- Stats Area: string (nullable = true)
 |-- Suburb: string (nullable = true)
 |-- Postcode: string (nullable = true)
 |-- LGA Name: string (nullable = true)
 |-- Total Units: string (nullable = true)
 |-- Total Cas: string (nullable = true)
 |-- Total Fats: string (nullable = true)
 |-- Total SI: string (nullable = true)
 |-- Total MI: string (nullable = true)
 |-- Year: string (nullable = true)
 |-- Month: string (nullable = true)
```

```
 |-- Day: string (nullable = true)
 |-- Time: string (nullable = true)
 |-- Area Speed: string (nullable = true)
 |-- Position Type: string (nullable = true)
 |-- Horizontal Align: string (nullable = true)
 |-- Vertical Align: string (nullable = true)
 |-- Other Feat: string (nullable = true)
 |-- Road Surface: string (nullable = true)
 |-- Moisture Cond: string (nullable = true)
 |-- Weather Cond: string (nullable = true)
 |-- DayNight: string (nullable = true)
 |-- Crash Type: string (nullable = true)
 |-- Unit Resp: string (nullable = true)
 |-- Entity Code: string (nullable = true)
 |-- CSEF Severity: string (nullable = true)
 |-- Traffic Ctrls: string (nullable = true)
 |-- DUI Involved: string (nullable = true)
 |-- Drugs Involved: string (nullable = true)
 |-- ACCLOC_X: string (nullable = true)
 |-- ACCLOC_Y: string (nullable = true)
 |-- UNIQUE_LOC: string (nullable = true)
```

2.2 Query/Analysis

2.2.1

In [15]:
```
from pyspark.sql.types import IntegerType
crash_df = crash_df.withColumn('Total Cas', crash_df['Total Cas'].c
ast(IntegerType()))
crash_ADE_df = crash_df.filter("Suburb = 'ADELAIDE'")
crash_ADE_df = crash_ADE_df.where(crash_ADE_df['Total Cas'] > 3)
crash_ADE_df.show()
```

```
+--------------------+----------+--------+--------+---------------
-+-----------+---------+----------+--------+--------+----+--------
+--------+--------+----------+-------------+----------------+-----
---------+--------------------+------------+-------------+--------
----+--------+--------------+---------+------------+-------------+
---------------+------------+--------------+----------+----------+
--------------+
|           REPORT_ID|Stats Area|  Suburb|Postcode|        LGA Nam
e|Total Units|Total Cas|Total Fats|Total SI|Total MI|Year|   Month
|     Day|    Time|Area Speed|Position Type|Horizontal Align|Verti
cal Align|          Other Feat|Road Surface|Moisture Cond|Weather
Cond|DayNight|    Crash Type|Unit Resp| Entity Code|CSEF Severity|
Traffic Ctrls|DUI Involved|Drugs Involved|  ACCLOC_X|  ACCLOC_Y|
UNIQUE_LOC|
+--------------------+----------+--------+--------+---------------
-+-----------+---------+----------+--------+--------+----+--------
+--------+--------+----------+-------------+----------------+-----
---------+--------------------+------------+-------------+--------
----+--------+--------------+---------+------------+-------------+
```

```
--------------+-----------+-------------+---------+---------+
-------------+
| 2018-601-17/01/2020|    1 City|ADELAIDE|    5000|CITY OF ADELAID
E|          8|        4|         0|       2|       2|2018| January
|  Sunday|09:12 pm|       050|  Not Divided|   Straight road|
Level|      Not Applicable|      Sealed|          Dry| Not Raining
|   Night|Hit Pedestrian|       01|Driver Rider|        3: SI|
No Control|        null|          null|1329806.36|1670224.76|13298
061670225|
|2017-1613-15/08/2019|    1 City|ADELAIDE|    5000|CITY OF ADELAID
E|          2|        4|         0|       0|       4|2017|February
|Saturday|04:00 pm|       050|   Cross Road|   Straight road|
Level|      Not Applicable|      Sealed|          Dry| Not Raining
|Daylight|    Right Turn|       01|Driver Rider|        2: MI|Traf
fic Signals|        null|          null|1327951.24|1669556.92|1327
9511669557|
|2017-12182-15/08/...|    1 City|ADELAIDE|    5000|CITY OF ADELAID
E|          6|        5|         0|       1|       4|2017|December
|Saturday|04:08 pm|       050|   Cross Road|   Straight road|
Level|      Not Applicable|      Sealed|          Wet| Not Raining
|Daylight|Hit Pedestrian|       01|Driver Rider|        3: SI|Traf
fic Signals|        null|          null| 1329016.2|1670995.07|1329
0161670995|
|2019-10404-8/07/2020|    1 City|ADELAIDE|    5000|CITY OF ADELAID
E|          4|        6|         0|       0|       6|2019| October
|  Monday|08:20 am|       060| Divided Road|   Straight road|
Level|Driveway or Entrance|      Sealed|          Dry| Not Raining
|Daylight|    Right Turn|       01|Driver Rider|        2: MI|
No Control|        null|          null|1327088.72|1670880.07|13270
891670880|
+--------------------+----------+--------+--------+--------------
-+-----------+---------+----------+--------+--------+----+--------
+--------+--------+----------+-------------+----------------+-----
---------+--------------------+------------+-------------+--------
----+--------+--------------+---------+------------+-------------+
---------------+------------+--------------+----------+----------+
--------------+
```

2.2.2

In [16]:
```
crash_top10_df = crash_df.orderBy('Total cas', ascending=False).lim
it(10)
crash_top10_df.orderBy('Total cas', ascending=False).show()
```

```
+--------------------+--------------+-------------+--------+------
--------------+-----------+---------+----------+--------+--------+
----+--------+---------+--------+----------+-------------+--------
------------+--------------+--------------+------------+----------
---+------------+--------+-----------+---------+------------+-----
--------+---------------+------------+--------------+----------+--
--------+--------------+
|           REPORT_ID|    Stats Area|       Suburb|Postcode|
LGA Name|Total Units|Total Cas|Total Fats|Total SI|Total MI|Year|
Month|      Day|    Time|Area Speed|Position Type|    Horizontal A
```

```
lign|Vertical Align|    Other Feat|Road Surface|Moisture Cond|Weat
her Cond|DayNight| Crash Type|Unit Resp| Entity Code|CSEF Severity
|  Traffic Ctrls|DUI Involved|Drugs Involved|  ACCLOC_X|  ACCLOC_Y
|    UNIQUE_LOC|
+-------------------+-------------+------------+--------+------
-------------+----------+---------+----------+--------+--------+
----+--------+---------+--------+----------+------------+--------
-----------+-------------+-------------+-----------+----------
---+-----------+--------+----------+---------+-----------+-----
--------+--------------+-----------+-------------+----------+--
--------+-------------+
| 2017-288-15/08/2019|2 Metropolitan|   PARA HILLS|    5096|   CIT
Y OF SALISBURY|          2|       11|         0|       1|      10|
2017| January|Wednesday|01:13 pm|       060|   T-Junction|       S
traight road| Crest of Hill|Not Applicable|      Sealed|
Dry| Not Raining|Daylight|Right Angle|       01|Driver Rider|
3: SI|      Stop Sign|        null|          null| 1334428.9|16830
32.96|13344291683033|
|2016-3035-15/08/2019|2 Metropolitan|      HACKHAM|    5163| CITY
OF ONKAPARINGA|          3|        9|         3|       5|       1|
2016| January| Saturday|11:50 am|       080|   T-Junction|       S
traight road|         Level|Not Applicable|      Sealed|
Dry| Not Raining|Daylight| Right Turn|       01|Driver Rider|
4: Fatal|     No Control|        null|          null|1320361.49|16
45195.63|13203611645196|
|2016-6630-15/08/2019|2 Metropolitan|KANGAROO FLAT|    5118|LIGHT
REGIONAL CO...|          3|        9|         0|       2|       7|
2016|   April|Wednesday|09:00 pm|       100|  Not Divided|CURVED,
VIEW OBSC...|         Level|Not Applicable|      Sealed|
Dry| Not Raining|   Night|    Head On|       01|Driver Rider|
3: SI|     No Control|        null|          null|1339316.32|17103
14.92|13393161710315|
|2019-11734-8/07/2020|2 Metropolitan|        STURT|    5047|CC MAR
ION.       ...|          2|        9|         0|       1|       8|
2019|November|   Sunday|07:25 pm|       060|   T-Junction|       S
traight road|         Level|Not Applicable|      Sealed|
Dry| Not Raining|Daylight| Right Turn|       02|Driver Rider|
3: SI|Traffic Signals|        null|          null|1324428.84|16598
84.95|13244291659885|
|2016-7073-15/08/2019|     3 Country|     MERRITON|    5523|PT.PIR
IE CITY & D...|          2|        8|         4|       3|       1|
2016|   April|   Sunday|12:35 pm|       110|  Not Divided|       S
traight road|         Level|Not Applicable|      Sealed|
Dry| Not Raining|Daylight|    Head On|       01|Driver Rider|
4: Fatal|     No Control|        null|          null|1293759.89|18
40109.96|12937601840110|
|2016-14407-15/08/...|     3 Country|    STOCKWELL|    5355|THE BA
ROSSA COUNCIL.|          2|        8|         1|       6|       1|
2016| October|   Sunday|03:46 pm|       100|  Not Divided|       S
traight road| Crest of Hill|Not Applicable|    Unsealed|
Dry| Not Raining|Daylight|    Head On|       01|Driver Rider|
4: Fatal|     No Control|        null|          null|1373964.45|17
23462.57|13739641723463|
|2015-2823-21/08/2019|     3 Country|       HAWKER|    5434|THE FL
INDERS RANG...|          1|        8|         0|       0|       8|
2015|   March|   Monday|06:00 pm|       110|  Not Divided|       S
traight road|         Level|Not Applicable|      Sealed|
```

```
Dry| Not Raining|Daylight|  Roll Over|       01|Driver Rider|
2: MI|     No Control|        null|          null|1315077.61|20223
09.34|13150782022309|
|2016-8547-15/08/2019|     3 Country|      WINDSOR|    5501|DC MAL
LALA.      ...|          4|        7|          0|       1|       6|
2016|     May| Saturday|03:45 pm|       110|  Not Divided|       S
traight road|         Level|Not Applicable|      Sealed|
Dry| Not Raining|Daylight|   Rear End|       02|Driver Rider|
3: SI|     No Control|        null|             Y|1306853.11|17249
52.66|13068531724953|
|2015-6965-21/08/2019|     3 Country|     BEAUFORT|    5550|YORKE
PENINSULA C...|          3|        7|          3|       4|       0|
2015|    June|   Monday|11:13 am|       100|   T-Junction|       S
traight road|         Level|Not Applicable|      Sealed|
Dry| Not Raining|Daylight|    Head On|       09|       Other|
4: Fatal|     No Control|        null|          null|1287930.19|17
61652.36|12879301761652|
|2015-12591-21/08/...|     3 Country|      MALLALA|    5502|DC MAL
LALA.      ...|          2|        7|          0|       2|       5|
2015| October|   Sunday|02:30 pm|       100|   Cross Road|       S
traight road|         Level|Not Applicable|    Unsealed|
Dry| Not Raining|Daylight|Right Angle|       01|Driver Rider|
3: SI|  Give Way Sign|        null|          null|1325122.01|17248
60.95|13251221724861|
+--------------------+--------------+-------------+--------+------
--------------+-----------+---------+----------+--------+--------+
----+--------+---------+--------+----------+-------------+--------
------------+--------------+--------------+------------+----------
---+------------+--------+-----------+---------+------------+-----
--------+---------------+------------+--------------+----------+--
--------+--------------+
```

2.2.3

In [17]:
```
crash_df = crash_df.withColumn('Total Fats', crash_df['Total Fats']
.cast(IntegerType()))

crash_type_cnt_df = crash_df.groupby('Crash Type').sum('Total Fats'
) \
                            .withColumnRenamed('sum(Total Fats)', 'Tota
l Fats')
crash_type_cnt_df.show()
```

```
+--------------------+----------+
|          Crash Type|Total Fats|
+--------------------+----------+
|           Roll Over|        57|
|  Hit Object on Road|         2|
|      Hit Pedestrian|        70|
|    Hit Fixed Object|       152|
|               Other|         2|
|          Side Swipe|        20|
|             Head On|        86|
|  Hit Parked Vehicle|         9|
|          Right Turn|        18|
|            Rear End|        16|
|          Hit Animal|         4|
|Left Road - Out o...|         1|
|         Right Angle|        45|
+--------------------+----------+
```

2.2.4

In [18]:
```
unlicensed_crash_df = unit_df.where(unit_df['Licence Type'] == 'Unl
icenced').select('REPORT_ID').distinct()
unlicensed_crash_df = crash_df.join(unlicensed_crash_df, crash_df.R
EPORT_ID == unlicensed_crash_df.REPORT_ID, \
                        how='left_semi').groupby('Suburb').sum('Tot
al Cas') \
                        .withColumnRenamed('sum(Total Cas)', 'Total
Casualties')
unlicensed_crash_df.show()
```

```
+------------------+----------------+
|            Suburb|Total Casualties|
+------------------+----------------+
|      FLINDERS PARK|               8|
|       POOGINAGORIC|               1|
|     TEA TREE GULLY|               1|
|            HACKHAM|               3|
|   MEDINDIE GARDENS|               0|
|           WISANGER|               1|
|            CUMMINS|               0|
|       BASKET RANGE|               0|
|MURRAY BRIDGE SOUTH|               0|
|      GILLES PLAINS|               7|
|            HAWKER|               0|
|           BEAUFORT|               1|
|             MAGILL|               7|
|            ECHUNGA|               1|
|            CULTANA|               1|
|        EDWARDSTOWN|               5|
|        RISDON PARK|               1|
|          THORNGATE|               1|
|       ANDREWS FARM|              12|
|       TORRENSVILLE|               5|
+------------------+----------------+
only showing top 20 rows
```

2.3 Severity Analysis

2.3.1

In [19]:
```
severity_cnt_df = crash_df.groupBy('CSEF Severity').count()
severity_cnt_df.show()
```

```
+-------------+-----+
|CSEF Severity|count|
+-------------+-----+
|     4: Fatal|  451|
|        2: MI|21881|
|       1: PDO|46696|
|        3: SI| 2978|
+-------------+-----+
```

Which severity level is the most common?

The property damage only (PDO) is the most common accident between 2015 and 2019, which happened 46,696 times.

2.3.2 a

In [20]:
```
total_DI_accident = crash_df.filter(crash_df['Drugs Involved'] =='Y
').count()
```

In [21]:
```
severity_cnt_DI_df = crash_df.filter(crash_df['Drugs Involved'] ==' 
Y') \
                     .groupBy('CSEF Severity').count()
severity_cnt_DI_df = severity_cnt_DI_df.withColumn('Percentage', se
verity_cnt_DI_df['count']/total_DI_accident * 100)
print('a. When the driver is tested positive on drugs.')
severity_cnt_DI_df.show()
```

```
a. When the driver is tested positive on drugs.
+-------------+-----+------------------+
|CSEF Severity|count|        Percentage|
+-------------+-----+------------------+
|     4: Fatal|   82| 6.539074960127592|
|        2: MI|  749|59.728867623604465|
|       1: PDO|  176|14.035087719298245|
|        3: SI|  247|19.696969696969695|
+-------------+-----+------------------+
```

2.3.2 b

In [22]:
```
total_DUI_accident = crash_df.filter(crash_df['DUI Involved'] =='Y'
).count()
```

In [23]:
```
severity_cnt_DUI_df = crash_df.filter(crash_df['DUI Involved'] =='Y
') \
                      .groupBy('CSEF Severity').count()
severity_cnt_DUI_df = severity_cnt_DUI_df.withColumn('Percentage',
severity_cnt_DUI_df['count']/total_DUI_accident * 100)

print('b. When the driver is tested positive for blood alcohol conc
entration.')
severity_cnt_DUI_df.show()
```

```
b. When the driver is tested positive for blood alcohol concentrat
ion.
+-------------+-----+------------------+
|CSEF Severity|count|        Percentage|
+-------------+-----+------------------+
|     4: Fatal|   79|  3.51423487544484|
|        2: MI|  737|  32.7846975088968|
|       1: PDO| 1173|52.179715302491104|
|        3: SI|  259| 11.52135231316726|
+-------------+-----+------------------+
```

2.3.2 c

In [24]:
```
total_DI_DUI_accident = crash_df.filter((crash_df['Drugs Involved']
=='Y') & (crash_df['DUI Involved'] =='Y')).count()
```

In [25]:
```
severity_cnt_DI_DUI_df = crash_df.filter((crash_df['Drugs Involved'
] =='Y') & (crash_df['DUI Involved'] =='Y')) \
                                  .groupBy('CSEF Severity').count()
severity_cnt_DI_DUI_df = severity_cnt_DI_DUI_df.withColumn('Percent
age', severity_cnt_DI_DUI_df['count']/total_DI_DUI_accident * 100)

print('c. When the driver is tested positive for both drugs and blo
od alcohol.')
severity_cnt_DI_DUI_df.show()
```

```
c. When the driver is tested positive for both drugs and blood alc
ohol.
+-------------+-----+------------------+
|CSEF Severity|count|        Percentage|
+-------------+-----+------------------+
|     4: Fatal|   27|15.428571428571427|
|        2: MI|   89|50.857142857142854|
|       1: PDO|   24|13.714285714285715|
|        3: SI|   35|              20.0|
+-------------+-----+------------------+
```

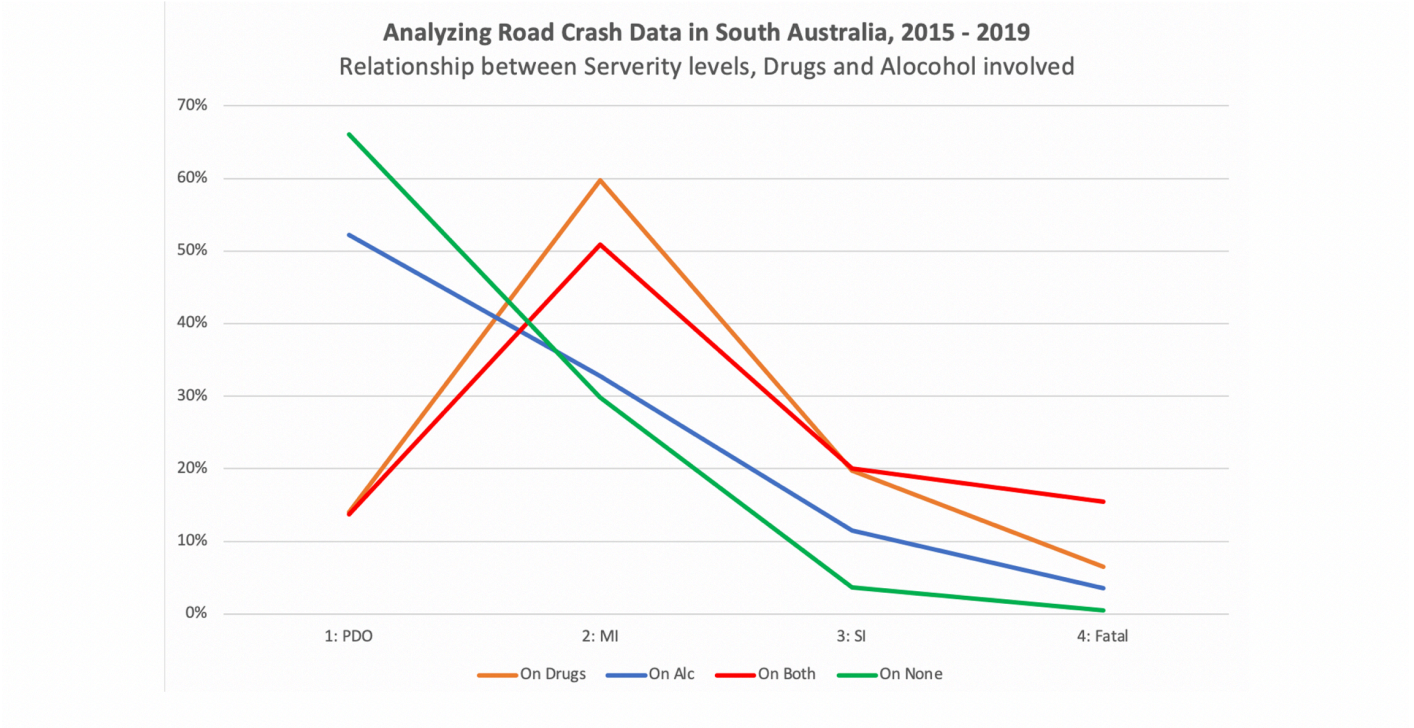
2.3.2 d

In [26]:
```
severity_cnt_no_DI_DUI_df = crash_df.fillna({'Drugs Involved': '',
'DUI Involved': ''})
severity_cnt_no_DI_DUI_df = severity_cnt_no_DI_DUI_df.filter((sever
ity_cnt_no_DI_DUI_df['Drugs Involved'] !='Y') \
                                                    & (seve
rity_cnt_no_DI_DUI_df['DUI Involved'] !='Y'))

total_no_DI_DUI_accident = severity_cnt_no_DI_DUI_df.count()
```

In [27]:
```
severity_cnt_no_DI_DUI_df = severity_cnt_no_DI_DUI_df.groupBy('CSEF
Severity').count()
severity_cnt_no_DI_DUI_df = severity_cnt_no_DI_DUI_df.withColumn('P
ercentage', severity_cnt_no_DI_DUI_df['count']/total_no_DI_DUI_acci
dent * 100)

print('d. When the driver is tested negative for both (no alcohol a
nd no drugs).')
severity_cnt_no_DI_DUI_df.show()
```

```
d. When the driver is tested negative for both (no alcohol and no
drugs).
+-------------+-----+------------------+
|CSEF Severity|count|        Percentage|
+-------------+-----+------------------+
|     4: Fatal|  317| 0.461567582521586|
|        2: MI|20484|29.825710916000524|
|       1: PDO|45371| 66.06240626683557|
|        3: SI| 2507|3.6503152346423215|
+-------------+-----+------------------+
```

Compare the results in these 4 scenarios and briefly explain the observation from this analysis.



Answer

Comparing the severity levels of crashes, it is obvious that On Drugs and On both are significantly high chance of getting badly damaged. Furthermore, On alcohol has slightly increased a severity level and lastly, On None of them is the most potentially safest in the accident.

2.4 Comparasion RDDs vs DataFrame vs Spark SQL

2.4.1

Using RDD

In [28]:
```
%%time
trnf_unit_rdd = unit_rdd.map(lambda x: x.split(",")) \
                    .map(lambda x: [s.strip("\"") for s in x]) \
                    .map(lambda x: (x[0], [x[7], x[8] , x[11]] ))

trnf_crash_rdd = crash_rdd.map(lambda x: x.split(",")) \
                    .map(lambda x: [s.strip("\"") for s in x]) \
                    .filter(lambda x: x[2].lower()=="adelaide") \
                    .map(lambda x: (x[0], [x[2] , x[6], x[10]+"-"+x
[11]+"-"+x[12] , x[13]]))

result_rdd = trnf_unit_rdd.join(trnf_crash_rdd)
result_rdd = result_rdd.map(lambda x: (x[1][1][2], x[1][1][3], x[1]
[1][1], x[1][1][0], \
                                x[1][0][0], x[1][0][1], x[1][0][2]))

print('Total records:', result_rdd.count(), "\n")
result_rdd.take(20)
```

```
Total records: 6310

CPU times: user 26.1 ms, sys: 5.61 ms, total: 31.7 ms
Wall time: 2.09 s
```

Out[28]:
```
[('2016-November-Wednesday',
  '04:26 pm',
  '0',
  'ADELAIDE',
  'Male',
  '017',
  'Unknown'),
 ('2016-November-Wednesday',
  '04:26 pm',
  '0',
  'ADELAIDE',
  'Male',
  '025',
  'Unknown'),
 ('2016-December-Friday', '11:30 am', '0', 'ADELAIDE', 'Male', '08
0', 'Full'),
 ('2016-December-Friday', '11:30 am', '0', 'ADELAIDE', 'Male', '04
8', 'Full'),
 ('2016-December-Saturday',
  '07:40 am',
  '0',
  'ADELAIDE',
  'Male',
  '032',
  'Full'),
 ('2016-December-Saturday',
  '07:40 am',
  '0',
  'ADELAIDE',
  'Unknown',
  'XXX',
  'Unknown'),
 ('2016-December-Friday',
```

```
  '05:30 pm',
  '0',
  'ADELAIDE',
  'Female',
  '058',
  'Full'),
 ('2016-December-Friday', '05:30 pm', '0', 'ADELAIDE', 'Male', '04
1', 'Full'),
 ('2016-December-Wednesday',
  '04:20 pm',
  '0',
  'ADELAIDE',
  'Female',
  '045',
  'Full'),
 ('2016-December-Wednesday',
  '04:20 pm',
  '0',
  'ADELAIDE',
  'Male',
  '027',
  'Full'),
 ('2016-December-Wednesday',
  '04:20 pm',
  '0',
  'ADELAIDE',
  'Male',
  '041',
  'Full'),
 ('2016-December-Thursday', '01:15 pm', '0', 'ADELAIDE', 'Unknown'
, 'XXX', ''),
 ('2016-December-Thursday',
  '01:15 pm',
  '0',
  'ADELAIDE',
  'Female',
  '021',
  'Full'),
 ('2016-December-Friday', '11:25 am', '0', 'ADELAIDE', 'Male', '02
3', 'Full'),
 ('2016-December-Friday',
  '11:25 am',
  '0',
  'ADELAIDE',
  'Unknown',
  'XXX',
  'Unknown'),
 ('2016-December-Monday',
  '02:50 pm',
  '0',
  'ADELAIDE',
  'Female',
  '048',
  'Full'),
 ('2016-December-Monday',
  '02:50 pm',
  '0',
```

```
  'ADELAIDE',
  'Unknown',
  'XXX',
  'Unknown'),
 ('2016-February-Sunday',
  '04:50 pm',
  '0',
  'ADELAIDE',
  'Female',
  '033',
  'Full'),
 ('2016-February-Sunday',
  '04:50 pm',
  '0',
  'ADELAIDE',
  'Male',
  'XXX',
  'Unknown'),
 ('2016-June-Monday', '06:50 pm', '0', 'ADELAIDE', 'Female', '022'
, 'Full')]
```

Using Dataframe

In [29]:

```
%%time
from pyspark.sql.functions import concat, lit, col

unit_filtered_df = unit_df.select(['REPORT_ID', 'Sex', 'Age', 'Lice
nce Type'])
crash_filtered_df = crash_df.filter(crash_df.Suburb == 'ADELAIDE')
\
                                .select(['REPORT_ID','Year','Month','Da
y','Time','Total Cas','Suburb'])

result_df = crash_filtered_df.join(unit_filtered_df, crash_filtered
_df.REPORT_ID == unit_filtered_df.REPORT_ID, how='inner')

print('Total records:', result_df.count(), "\n")
result_df.select([concat(result_df['Year'],lit('-'),result_df['Mont
h'],lit('-'),result_df['Day']).alias("Date") \
                  ,'Time',result_df['Total Cas'].alias('Number of C
asualties') \
                  ,'Suburb',result_df['Sex'].alias('Gender'), 'Age'
, 'Licence Type']).show(20)
```

Total records: 6310

Date	Time	Number of Casualties	Suburb	Gender	Age	Licence Type
2016-November-Wed...	01:45 pm	1	ADELAIDE	Male	056	Full
2016-November-Wed...	01:45 pm	1	ADELAIDE	Male	072	null
2016-November-Tue...	03:40 pm	1	ADELAIDE	Male	056	null
2016-November-Tue...	03:40 pm	1	ADELAIDE	Female	027	null
2016-November-Tue...	05:00 pm	0	ADELAIDE	Female	032	Full
2016-November-Tue...	05:00 pm	0	ADELAIDE	Unknown	XXX	Unknown
2016-November-Tue...	05:40 pm	0	ADELAIDE	Male	022	Unknown
2016-November-Tue...	05:40 pm	0	ADELAIDE	Male	020	Unknown
2016-November-Monday	11:26 pm	0	ADELAIDE	Unknown	XXX	Unknown
2016-November-Monday	11:26 pm	0	ADELAIDE	Male	042	Full
2016-November-Monday	11:26 pm	0	ADELAIDE	null	null	null
2016-November-Monday	11:30 pm	0	ADELAIDE	Male	026	Unknown
2016-November-Monday	11:30 pm	0	ADELAIDE	Male	038	Full
2016-November-Monday	11:30 pm	0	ADELAIDE	Male	036	Full
2016-November-Tue...	05:05 pm	0	ADELAIDE	Male	025	Unknown
2016-November-Tue...	05:05 pm	0	ADELAIDE	Unknown	XXX	Unknown
2016-November-Wed...	03:30 pm	1	ADELAIDE	Female	065	Full
2016-November-Wed...	03:30 pm	1	ADELAIDE	Unknown	XXX	Unknown
2016-November-Wed...	02:20 pm	0	ADELAIDE	Male	063	Full
2016-November-Wed...	02:20 pm	0	ADELAIDE	null	null	null

only showing top 20 rows

CPU times: user 16.2 ms, sys: 923 μs, total: 17.1 ms
Wall time: 1.1 s

Using SparkSQL

In [30]:
```
# Create Views from Dataframes
unit_df.createOrReplaceTempView("sql_unit")
crash_df.createOrReplaceTempView("sql_crash")
```

In [31]:
```
%%time

## select data using SQL
sql_result = spark.sql('''
  SELECT CONCAT(c.Year,'-',c.Month,'-',c.Day) `Date`, c.Time , `Tot
al cas` `Number of Casualties`,c.Suburb, u.Sex Gender, u.Age, `Lice
nce Type`
  FROM sql_unit u JOIN sql_crash c
  ON u.REPORT_ID = c.REPORT_ID
  WHERE c.Suburb = 'ADELAIDE'
''')

print('Total records:', sql_result.count(), "\n")
sql_result.show(20)
```

```
Total records: 6310

+--------------------+--------+--------------------+--------+-----
--+----+------------+
|                Date|    Time|Number of Casualties|  Suburb| Gend
er| Age|Licence Type|
+--------------------+--------+--------------------+--------+-----
--+----+------------+
|2016-November-Wed...|01:45 pm|                   1|ADELAIDE|   Ma
le| 056|        Full|
|2016-November-Wed...|01:45 pm|                   1|ADELAIDE|   Ma
le| 072|        null|
|2016-November-Tue...|03:40 pm|                   1|ADELAIDE|   Ma
le| 056|        null|
|2016-November-Tue...|03:40 pm|                   1|ADELAIDE| Fema
le| 027|        null|
|2016-November-Tue...|05:00 pm|                   0|ADELAIDE| Fema
le| 032|        Full|
|2016-November-Tue...|05:00 pm|                   0|ADELAIDE|Unkno
wn| XXX|     Unknown|
|2016-November-Tue...|05:40 pm|                   0|ADELAIDE|   Ma
le| 022|     Unknown|
|2016-November-Tue...|05:40 pm|                   0|ADELAIDE|   Ma
le| 020|     Unknown|
|2016-November-Monday|11:26 pm|                   0|ADELAIDE|Unkno
wn| XXX|     Unknown|
|2016-November-Monday|11:26 pm|                   0|ADELAIDE|   Ma
le| 042|        Full|
|2016-November-Monday|11:26 pm|                   0|ADELAIDE|   nu
ll|null|        null|
|2016-November-Monday|11:30 pm|                   0|ADELAIDE|   Ma
le| 026|     Unknown|
|2016-November-Monday|11:30 pm|                   0|ADELAIDE|   Ma
le| 038|        Full|
|2016-November-Monday|11:30 pm|                   0|ADELAIDE|   Ma
le| 036|        Full|
|2016-November-Tue...|05:05 pm|                   0|ADELAIDE|   Ma
le| 025|     Unknown|
|2016-November-Tue...|05:05 pm|                   0|ADELAIDE|Unkno
wn| XXX|     Unknown|
|2016-November-Wed...|03:30 pm|                   1|ADELAIDE| Fema
le| 065|        Full|
|2016-November-Wed...|03:30 pm|                   1|ADELAIDE|Unkno
wn| XXX|     Unknown|
|2016-November-Wed...|02:20 pm|                   0|ADELAIDE|   Ma
le| 063|        Full|
|2016-November-Wed...|02:20 pm|                   0|ADELAIDE|   nu
ll|null|        null|
+--------------------+--------+--------------------+--------+-----
--+----+------------+
only showing top 20 rows

CPU times: user 2.32 ms, sys: 654 µs, total: 2.97 ms
Wall time: 981 ms
```

2.4.2

Using RDD

In [32]:
```
%%time
trnf_unit_rdd = unit_rdd.map(lambda x: x.split(",")) \
                    .map(lambda x: (x[0], x[11] )) \
                    .map(lambda x: [s.strip("\"") for s in x]) \
                    .filter(lambda x: x[1].lower()=="unlicenced")

trnf_crash_rdd = crash_rdd.map(lambda x: x.split(",")) \
                    .map(lambda x: [s.strip("\"") for s in x]) \
                    .map(lambda x: (x[0], [x[2], int(x[6])]))

result_rdd = trnf_crash_rdd.join(trnf_unit_rdd).map(lambda x: (x[1]
[0][0],x[1][0][1]))
result_rdd = result_rdd.reduceByKey(lambda a,b: a+b)

print('Total records:', result_rdd.count(), "\n")
result_rdd.take(20)
```

```
Total records: 634

CPU times: user 28.4 ms, sys: 0 ns, total: 28.4 ms
Wall time: 1.74 s
```

Out[32]:
```
[('ONKAPARINGA HILLS', 7),
 ('FIRLE', 6),
 ('RIVERGLADES', 1),
 ('SALISBURY PARK', 0),
 ('HAPPY VALLEY', 5),
 ('MUNNO PARA WEST', 1),
 ('HEATHPOOL', 3),
 ('NORTH PLYMPTON', 1),
 ('WHITES FLAT', 0),
 ('CRAIGMORE', 2),
 ('URRBRAE', 2),
 ('BIRKENHEAD', 2),
 ('PARALOWIE', 4),
 ('BERRI', 3),
 ('CLOVELLY PARK', 1),
 ('DAVOREN PARK', 8),
 ('HACKHAM', 3),
 ('MODBURY NORTH', 3),
 ('FULLARTON', 0),
 ('ASCOT PARK', 2)]
```

Using DataFrame

In [33]:
```
%%time

unit_filtered_df = unit_df.where(unit_df['Licence Type'] == 'Unlice
nced').select('REPORT_ID')
result_df = crash_df.join(unit_filtered_df, crash_df.REPORT_ID == u
nit_filtered_df.REPORT_ID, \
                          how='left_semi').groupby('Suburb').sum('Tot
al Cas') \
                          .withColumnRenamed('sum(Total Cas)', 'Total
Casualties')

print('Total records:', result_df.count(), "\n")
result_df.show()
```

```
Total records: 634

+------------------+----------------+
|            Suburb|Total Casualties|
+------------------+----------------+
|      FLINDERS PARK|               8|
|       POOGINAGORIC|               1|
|     TEA TREE GULLY|               1|
|            HACKHAM|               3|
|   MEDINDIE GARDENS|               0|
|           WISANGER|               1|
|            CUMMINS|               0|
|       BASKET RANGE|               0|
|MURRAY BRIDGE SOUTH|               0|
|      GILLES PLAINS|               7|
|             HAWKER|               0|
|           BEAUFORT|               1|
|             MAGILL|               7|
|            ECHUNGA|               1|
|            CULTANA|               1|
|        EDWARDSTOWN|               5|
|        RISDON PARK|               1|
|          THORNGATE|               1|
|       ANDREWS FARM|              12|
|       TORRENSVILLE|               5|
+------------------+----------------+
only showing top 20 rows

CPU times: user 4.95 ms, sys: 1.37 ms, total: 6.32 ms
Wall time: 2.19 s
```

Using Spark SQL

In [34]:
```
%%time
## select data using SQL
sql_result = spark.sql('''
    SELECT c.suburb, sum(c.`Total cas`)
    FROM sql_crash c
          inner join sql_unit u ON u.REPORT_ID = c.REPORT_ID
    WHERE u.`Licence Type` = 'Unlicenced'
    GROUP BY c.suburb
''')

print('Total records:', sql_result.count(), "\n")
sql_result.show(20)
```

```
Total records: 634

+------------------+--------------+
|            suburb|sum(Total cas)|
+------------------+--------------+
|      FLINDERS PARK|             8|
|       POOGINAGORIC|             1|
|     TEA TREE GULLY|             1|
|            HACKHAM|             3|
|   MEDINDIE GARDENS|             0|
|           WISANGER|             1|
|            CUMMINS|             0|
|       BASKET RANGE|             0|
|MURRAY BRIDGE SOUTH|             0|
|      GILLES PLAINS|             7|
|             HAWKER|             0|
|           BEAUFORT|             1|
|             MAGILL|             7|
|            ECHUNGA|             1|
|            CULTANA|             1|
|        EDWARDSTOWN|             6|
|        RISDON PARK|             1|
|          THORNGATE|             1|
|       ANDREWS FARM|            12|
|       TORRENSVILLE|             5|
+------------------+--------------+
only showing top 20 rows

CPU times: user 3.69 ms, sys: 0 ns, total: 3.69 ms
Wall time: 2.03 s
```

In [ ]: