> **Algoritmos para Big Data**
>
> **2023/24**

Exploratory Data Analysis (EDA)

In this lecture, we will carry out some common data analysis to extract insights from data stored in a DataFrame. The main goal is to consolidate knownledge from previous lectures about useful Spark's transformations and actions functions.

Problem formulation

This exercise is about EDA related to **Fire Department calls for service** in San Francisco, USA.

We ask you write down a Spark program that:

a) Reads a file containing the dataset under analysis.

b) Provides answers to the following questions about the data.

1. How many distinct types of calls were made to the Fire Department?
2. What are distinct types of calls that were made to the Fire Department?
3. Find out all responses or delayed times that were greater than 5 minutes?
4. What were the most common call types, listed in descending order by count?
5. What zip codes accounted for most common calls and what type were they?
6. What neighbourhoods are in the two top zip codes from the listing in the previous question?

Dataset

The dataset of concern, and related information, can be downloaded using the command

> wget bigdata.iscte.me/abd/fire-department-calls.zip

Initial settings

Prior to any computation, let us set required imports and create a Spark session, as well as defining useful functions.

In [1]:
```
import findspark, pyspark

from pyspark.sql import SparkSession
from pyspark.sql.types import *
import pyspark.sql.functions as F
```

In [2]:
```
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

In [3]:
```
# Create our Spark session

findspark.init()
findspark.find()

spark = SparkSession\
        .builder\
        .appName("FireDepartmentCalls")\
        .config("spark.sql.shuffle.partitions",6)\
        .config("spark.sql.repl.eagereval.enabled",True)\
        .getOrCreate()
```

```
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
24/03/05 14:29:30 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-
```

In [4]:
```
from IPython.core.display import HTML
display(HTML("<style>pre { white-space: pre !important; }</style>"))
```

Useful visualization function

Function relying on Seaborn to plot data but as Python data frame.

See https://seaborn.pydata.org/index.html

We encourage you to use your own plotting functions. Remember: *"A picture is worth a thousand words"*

In [5]:
```
def plotHorizBar(df, xcol, ycol, colour):
    sns.barplot(data=df, x=xcol, y=ycol, color=colour)
```

Data ingestion

Checking source of data and then reading it.

In [ ]:
```
pwd
```

In [ ]:
```
ls -la
```

In [ ]:
```
ls -la ../Datasets/fire-department-calls
```

In [9]:
```
! head -n 2 "../Datasets/fire-department-calls/Fire_Department_Calls_for_Service_Excel_EU.csv"
```

```
Call Number;Unit ID;Incident Number;Call Type;Call Date;Watch Date;Received DtTm;Entry DtTm;Dispatch DtTm;Respons
221210313;E36;22054955;Outside Fire;05/01/2022;04/30/2022;05/01/2022 02:58:25 AM;05/01/2022 02:59:15 AM;05/01/202
```

In [10]:

```
# As the file is quite big, (recall that inferring the schema is expensive for large files)
# and we know it, so let us use it (How do we know it?)

fire_schema = StructType([StructField('Call Number', IntegerType(), True),
                     StructField('Unit ID', StringType(), True),
                     StructField('Incident Number', IntegerType(), True),
                     StructField('Call Type', StringType(), True),
                     StructField('Call Date', StringType(), True),
                     StructField('Watch Date', StringType(), True),
                     StructField('Received DtTm', StringType(), True),
                     StructField('Entry DtTm', StringType(), True),
                     StructField('Dispatch DtTm', StringType(), True),
                     StructField('Response DtTm', StringType(), True),
                     StructField('On Scene DtTm', StringType(), True),
                     StructField('Transport DtTm', StringType(), True),
                     StructField('Hospital DtTm', StringType(), True),
                     StructField('Call Final Disposition', StringType(), True),
                     StructField('Available DtTm', StringType(), True),
                     StructField('Address', StringType(), True),
                     StructField('City', StringType(), True),
                     StructField('Zipcode of Incident', IntegerType(), True),
                     StructField('Battalion', StringType(), True),
                     StructField('Station Area', StringType(), True),
                     StructField('Box', StringType(), True),
                     StructField('Original Priority', StringType(), True),
                     StructField('Priority', StringType(), True),
                     StructField('Final Priority', IntegerType(), True),
                     StructField('ALS Unit', BooleanType(), True),
                     StructField('Call Type Group', StringType(), True),
                     StructField('Number of Alarms', IntegerType(), True),
                     StructField('Unit Type', StringType(), True),
                     StructField('Unit sequence in call dispatch', IntegerType(), True),
                     StructField('Fire Prevention District', StringType(), True),
                     StructField('Supervisor District', StringType(), True),
                     StructField('Neighborhooods - Analysis Boundaries', StringType(), True),
                     StructField('RowID', StringType(), True),
                     StructField('case_location', StringType(), True),
                     StructField('Analysis Neighborhoods', IntegerType(), True)])
```

In [11]:
```
# Reading the dataset

filename =

fire_df = spark.read.csv(filename
```

Checking data

Schema, show, count, and statistical information.

In [12]:
```
# Schema

fire_df.
```

```
root
 |-- Call Number: integer (nullable = true)
 |-- Unit ID: string (nullable = true)
 |-- Incident Number: integer (nullable = true)
 |-- Call Type: string (nullable = true)
 |-- Call Date: string (nullable = true)
 |-- Watch Date: string (nullable = true)
 |-- Received DtTm: string (nullable = true)
 |-- Entry DtTm: string (nullable = true)
 |-- Dispatch DtTm: string (nullable = true)
 |-- Response DtTm: string (nullable = true)
 |-- On Scene DtTm: string (nullable = true)
 |-- Transport DtTm: string (nullable = true)
 |-- Hospital DtTm: string (nullable = true)
 |-- Call Final Disposition: string (nullable = true)
 |-- Available DtTm: string (nullable = true)
 |-- Address: string (nullable = true)
 |-- City: string (nullable = true)
 |-- Zipcode of Incident: integer (nullable = true)
 |-- Battalion: string (nullable = true)
 |-- Station Area: string (nullable = true)
 |-- Box: string (nullable = true)
 |-- Original Priority: string (nullable = true)
 |-- Priority: string (nullable = true)
 |-- Final Priority: integer (nullable = true)
 |-- ALS Unit: boolean (nullable = true)
 |-- Call Type Group: string (nullable = true)
 |-- Number of Alarms: integer (nullable = true)
 |-- Unit Type: string (nullable = true)
 |-- Unit sequence in call dispatch: integer (nullable = true)
 |-- Fire Prevention District: string (nullable = true)
 |-- Supervisor District: string (nullable = true)
 |-- Neighborhooods - Analysis Boundaries: string (nullable = true)
 |-- RowID: string (nullable = true)
 |-- case_location: string (nullable = true)
 |-- Analysis Neighborhoods: integer (nullable = true)
```

In [13]:
```
# Show

fire_df.
```

```
24/03/05 14:29:40 WARN SparkStringUtils: Truncated the string representation of a plan since it was too large. Th
+-----------+-------+---------------+------------+----------+----------+--------------------+--------------------
|Call Number|Unit ID|Incident Number|   Call Type| Call Date|Watch Date|       Received DtTm|          Entry DtTm
+-----------+-------+---------------+------------+----------+----------+--------------------+--------------------
|  221210313|    E36|       22054955|Outside Fire|05/01/2022|04/30/2022|05/01/2022 02:58:...|05/01/2022 02:59:...
|  220190150|    E29|       22008871|      Alarms|01/19/2022|01/18/2022|01/19/2022 01:42:...|01/19/2022 01:44:...
|  211233271|    T07|       21053032|      Alarms|05/03/2021|05/03/2021|05/03/2021 09:28:...|05/03/2021 09:28:...
|  212933533|    B02|       21127914|      Alarms|10/20/2021|10/20/2021|10/20/2021 10:08:...|10/20/2021 10:09:...
|  221202543|    E41|       22054815|      Alarms|04/30/2022|04/30/2022|04/30/2022 06:35:...|04/30/2022 06:37:...
+-----------+-------+---------------+------------+----------+----------+--------------------+--------------------
only showing top 5 rows
```

In [14]:
```
# Count

fire_df
```

Out[14]:
```
6106908
```

Exploratory data analysis

Prior to any further analysis, we should consider upating the data types of time related fields. The data dictionary provided shows the fields that are considered as of *Date & Time*.

The default format should be MM-dd-yyyy HH:mm:ss.SSS ...

but it looks like we have MM/dd/yyyy HH:mm:ss PM (or AM)

Column	Type	Description
Call Date	Date & Time	Date the call is received at the 911 Dispatch Center. Used for reporting purposes.
Watch Date	Date & Time	Watch date when the call is received. Watch date starts at 0800 each morning and ends at 0800 the next day.
Received DtTm	Date & Time	Date and time of call is received at the 911 Dispatch Center.
Entry DtTm	Date & Time	Date and time the 911 operator submits the entry of the initical call information into the CAD system.
Dispatch DtTm	Date & Time	Date and time the 911 operator dispatches this unit to the call.
Response DtTm	Date & Time	Date and time this unit acknowledges the dispatch and records that the unit is en route to the location of the call.
On Scene DtTm	Date & Time	Date and time the unit records arriving to the location of the incident.
Transport DtTm	Date & Time	If this unit is an ambulance, date and time the unit begins the transport unit arrives to hospital.
Hospital DtTm	Date & Time	If this unit is an ambulance, date and time the unit arrives to the hospital.
Available DtTm	Date & Time	Date and time this unit is not longer assigned to this call and it is available for another dispatch.

In [15]:
```
# See https://spark.apache.org/docs/latest/sql-ref-datetime-pattern.html

new_fire_df = ( fire_df
                .withColumn("Call Date ts", F.to_timestamp("Call Date","MM/dd/yyyy"))
                .withColumn("Watch Date ts", F.to_timestamp("Watch Date","MM/dd/yyyy"))
                .withColumn("Received DtTm ts", F.to_timestamp("Received DtTm","MM/dd/yyyy KK:mm:ss a"))
                .withColumn("Entry DtTm ts", F.to_timestamp("Entry DtTm","MM/dd/yyyy KK:mm:ss a"))
                .withColumn("Dispatch DtTm ts", F.to_timestamp("Dispatch DtTm","MM/dd/yyyy KK:mm:ss a"))
                .withColumn("Response DtTm ts", F.to_timestamp("Response DtTm","MM/dd/yyyy KK:mm:ss a"))
                .withColumn("On Scene DtTm ts", F.to_timestamp("On Scene DtTm","MM/dd/yyyy KK:mm:ss a"))
                .withColumn("Transport DtTm ts", F.to_timestamp("Transport DtTm","MM/dd/yyyy KK:mm:ss a"))
                .withColumn("Entry DtTm ts", F.to_timestamp("Entry DtTm","MM/dd/yyyy KK:mm:ss a"))
                .withColumn("Hospital DtTm ts", F.to_timestamp("Hospital DtTm","MM/dd/yyyy KK:mm:ss a"))
                .withColumn("Available DtTm ts", F.to_timestamp("Available DtTm","MM/dd/yyyy KK:mm:ss a"))
                )
```

In [16]:
```
# Check changes

new_fire_df.
new_fire_df.
```

```
root
 |-- Call Number: integer (nullable = true)
 |-- Unit ID: string (nullable = true)
 |-- Incident Number: integer (nullable = true)
 |-- Call Type: string (nullable = true)
 |-- Call Date: string (nullable = true)
 |-- Watch Date: string (nullable = true)
 |-- Received DtTm: string (nullable = true)
 |-- Entry DtTm: string (nullable = true)
 |-- Dispatch DtTm: string (nullable = true)
 |-- Response DtTm: string (nullable = true)
 |-- On Scene DtTm: string (nullable = true)
 |-- Transport DtTm: string (nullable = true)
 |-- Hospital DtTm: string (nullable = true)
 |-- Call Final Disposition: string (nullable = true)
 |-- Available DtTm: string (nullable = true)
 |-- Address: string (nullable = true)
 |-- City: string (nullable = true)
 |-- Zipcode of Incident: integer (nullable = true)
 |-- Battalion: string (nullable = true)
```

```
 |-- Station Area: string (nullable = true)
 |-- Box: string (nullable = true)
 |-- Original Priority: string (nullable = true)
 |-- Priority: string (nullable = true)
 |-- Final Priority: integer (nullable = true)
 |-- ALS Unit: boolean (nullable = true)
 |-- Call Type Group: string (nullable = true)
 |-- Number of Alarms: integer (nullable = true)
 |-- Unit Type: string (nullable = true)
 |-- Unit sequence in call dispatch: integer (nullable = true)
 |-- Fire Prevention District: string (nullable = true)
 |-- Supervisor District: string (nullable = true)
 |-- Neighborhooods - Analysis Boundaries: string (nullable = true)
 |-- RowID: string (nullable = true)
 |-- case_location: string (nullable = true)
 |-- Analysis Neighborhoods: integer (nullable = true)
 |-- Call Date ts: timestamp (nullable = true)
 |-- Watch Date ts: timestamp (nullable = true)
 |-- Received DtTm ts: timestamp (nullable = true)
 |-- Entry DtTm ts: timestamp (nullable = true)
 |-- Dispatch DtTm ts: timestamp (nullable = true)
 |-- Response DtTm ts: timestamp (nullable = true)
 |-- On Scene DtTm ts: timestamp (nullable = true)
 |-- Transport DtTm ts: timestamp (nullable = true)
 |-- Hospital DtTm ts: timestamp (nullable = true)
 |-- Available DtTm ts: timestamp (nullable = true)
```

Call Number	Unit ID	Incident Number	Call Type	Call Date	Watch Date	Received DtTm	En
221210313	E36	22054955	Outside Fire	05/01/2022	04/30/2022	05/01/2022 02:58:...	05/01/2022 0
220190150	E29	22008871	Alarms	01/19/2022	01/18/2022	01/19/2022 01:42:...	01/19/2022 0
211233271	T07	21053032	Alarms	05/03/2021	05/03/2021	05/03/2021 09:28:...	05/03/2021 0
212933533	B02	21127914	Alarms	10/20/2021	10/20/2021	10/20/2021 10:08:...	10/20/2021 1
221202543	E41	22054815	Alarms	04/30/2022	04/30/2022	04/30/2022 06:35:...	04/30/2022 0
211232439	B01	21052945	Alarms	05/03/2021	05/03/2021	05/03/2021 04:57:...	05/03/2021 0
211942517	T03	21083057	Alarms	07/13/2021	07/13/2021	07/13/2021 04:50:...	07/13/2021 0
212932758	B01	21127810	Alarms	10/20/2021	10/20/2021	10/20/2021 05:46:...	10/20/2021 0
221201816	T03	22054719	Structure Fire	04/30/2022	04/30/2022	04/30/2022 02:27:...	04/30/2022 0
211941580	SCRT4	21082970	Medical Incident	07/13/2021	07/13/2021	07/13/2021 12:23:...	07/13/2021 1
220181779	50	22008631	Other	01/18/2022	01/18/2022	01/18/2022 01:38:...	01/18/2022 0

```
|  220111608|    E06|       22005327|              Alarms|01/11/2022|01/11/2022|01/11/2022 01:05:...|01/11/2022 0
|  220111597|  AM110|       22005326|    Medical Incident|01/11/2022|01/11/2022|01/11/2022 12:59:...|01/11/2022 0
|  220111595|    E07|       22005325|        Outside Fire|01/11/2022|01/11/2022|01/11/2022 01:01:...|01/11/2022 0
|  220181524|    E08|       22008605|              Alarms|01/18/2022|01/18/2022|01/18/2022 12:19:...|01/18/2022 1
|  221201435|    E41|       22054664|        Outside Fire|04/30/2022|04/30/2022|04/30/2022 12:32:...|04/30/2022 1
|  211941035|    T05|       21082896|Citizen Assist / ...|07/13/2021|07/13/2021|07/13/2021 10:00:...|07/13/2021 1
|  220181030|    B10|       22008545|              Alarms|01/18/2022|01/18/2022|01/18/2022 10:09:...|01/18/2022 1
|  222091290|    T10|       22096127|              Alarms|07/28/2022|07/28/2022|07/28/2022 11:47:...|07/28/2022 1
|  222091252|  SCRT3|       22096119|               Other|07/28/2022|07/28/2022|07/28/2022 11:39:...|07/28/2022 1
+-----------+-------+---------------+--------------------+----------+----------+--------------------+-----------
only showing top 20 rows
```

In [17]:
```
# Delete old dataframe if no longer needed

del fire_df
```

In [18]:
```
# Cache the DataFrame since we will be performing many operations on it.
# It makes operations faster at expenses of memory storage.
# Or better not using it!

# new_fire_df.cache()
```

Questions to be anwsered

In [19]:
```
# Just to review columns' name

new_fire_df.
```

Out[19]:
```
['Call Number',
 'Unit ID',
 'Incident Number',
 'Call Type',
 'Call Date',
 'Watch Date',
 'Received DtTm',
 'Entry DtTm',
 'Dispatch DtTm',
```

```
'Response DtTm',
'On Scene DtTm',
'Transport DtTm',
'Hospital DtTm',
'Call Final Disposition',
'Available DtTm',
'Address',
'City',
'Zipcode of Incident',
'Battalion',
'Station Area',
'Box',
'Original Priority',
'Priority',
'Final Priority',
'ALS Unit',
'Call Type Group',
'Number of Alarms',
'Unit Type',
'Unit sequence in call dispatch',
'Fire Prevention District',
'Supervisor District',
'Neighborhooods - Analysis Boundaries',
'RowID',
'case_location',
'Analysis Neighborhoods',
'Call Date ts',
'Watch Date ts',
'Received DtTm ts',
'Entry DtTm ts',
'Dispatch DtTm ts',
'Response DtTm ts',
'On Scene DtTm ts',
'Transport DtTm ts',
'Hospital DtTm ts',
'Available DtTm ts']
```

In [20]:
```
# Set a short list of main columns just for showing purposes, if needed

main_cols = ['Call Number',
 'Unit ID',
 'Incident Number',
 'Call Type',
 'Call Final Disposition',
 'Available DtTm',
 'Address',
 'City',
 'Zipcode of Incident',
 'Call Date ts',
 'Watch Date ts',
 'Received DtTm ts',
 'Entry DtTm ts',
 'Dispatch DtTm ts',
 'Response DtTm ts',
 'On Scene DtTm ts',
 'Transport DtTm ts',
 'Hospital DtTm ts',
 'Available DtTm ts']
```

1) How many distinct types of calls were made to the Fire Department?

Of course, we will not count "null" strings in that column.

In [21]:
```
new_fire_df.
```

Out[21]:
```
33
```

2) What are the distinct types of calls that were made to the Fire Department?

In [22]:
```
new_fire_df.
```

```
[Stage 13:=====================================================>   (17 + 1) / 18]
```

Call Type
Electrical Hazard
High Angle Rescue
Assist Police
Train / Rail Incident
Medical Incident
Vehicle Fire
Explosion
Confined Space / Structure Collapse
Industrial Accidents
Administrative
Train / Rail Fire
Alarms
Structure Fire
Water Rescue
Elevator / Escalator Rescue
Smoke Investigation (Outside)
HazMat
Marine Fire
Outside Fire
Citizen Assist / Service Call
Odor (Strange / Unknown)
Gas Leak (Natural and LP Gases)
Mutual Aid / Assist Outside Agency
Extrication / Entrapped (Machinery, Vehicle)
Other
Traffic Collision
Fuel Spill
Watercraft in Distress
Suspicious Package
Oil Spill
Aircraft Emergency
Structure Fire / Smoke in Building
Lightning Strike (Investigation)

3) Find out all responses or delayed times that were greater than 5 minutes?

(from the moment call is received till response is acknowledged and unit is on route)

1. Creates a new field *Response Delay* with the delay in minutes
2. Filter out the records with delay higher than 5 minutes.

In [23]:
```
new_fire_df = new_fire_df.withColumn("Response Delay",
        F.unix_timestamp(F.col("Response DtTm ts")) - F.unix_timestamp(F.col("Received DtTm ts")))
```

In [24]:
```
cols_to_show = main_cols + ['Response Delay']
new_fire_df.
```

```
+-----------+-------+---------------+-----------------------------+----------------------+----------------------+
|Call Number|Unit ID|Incident Number|Call Type                    |Call Final Disposition|Available DtTm        |
+-----------+-------+---------------+-----------------------------+----------------------+----------------------+
|221210313  |E36    |22054955       |Outside Fire                 |Fire                  |05/01/2022 03:05:00 AM|
|220190150  |E29    |22008871       |Alarms                       |Fire                  |01/19/2022 02:35:26 AM|
|211233271  |T07    |21053032       |Alarms                       |Fire                  |05/03/2021 09:38:09 PM|
|212933533  |B02    |21127914       |Alarms                       |Fire                  |10/20/2021 10:25:52 PM|
|221202543  |E41    |22054815       |Alarms                       |Fire                  |04/30/2022 06:40:08 PM|
|211232439  |B01    |21052945       |Alarms                       |Fire                  |05/03/2021 05:05:20 PM|
|211942517  |T03    |21083057       |Alarms                       |Fire                  |07/13/2021 04:54:45 PM|
|212932758  |B01    |21127810       |Alarms                       |Fire                  |10/20/2021 06:00:04 PM|
|221201816  |T03    |22054719       |Structure Fire               |Fire                  |04/30/2022 02:46:15 PM|
|211941580  |SCRT4  |21082970       |Medical Incident             |SFPD                  |07/13/2021 12:48:46 PM|
|220181779  |50     |22008631       |Other                        |Fire                  |01/18/2022 03:55:14 PM|
|220111608  |E06    |22005327       |Alarms                       |Fire                  |01/11/2022 01:16:05 PM|
|220111597  |AM110  |22005326       |Medical Incident             |Code 2 Transport      |01/11/2022 03:36:28 PM|
|220111595  |E07    |22005325       |Outside Fire                 |Fire                  |01/11/2022 01:07:41 PM|
|220181524  |E08    |22008605       |Alarms                       |Fire                  |01/18/2022 12:27:19 PM|
|221201435  |E41    |22054664       |Outside Fire                 |Fire                  |04/30/2022 12:41:42 PM|
|211941035  |T05    |21082896       |Citizen Assist / Service Call|Fire                  |07/13/2021 10:10:08 AM|
|220181030  |B10    |22008545       |Alarms                       |Fire                  |01/18/2022 10:21:57 AM|
|222091290  |T10    |22096127       |Alarms                       |Fire                  |07/28/2022 11:56:01 AM|
|222091252  |SCRT3  |22096119       |Other                        |Cancelled             |07/28/2022 11:40:38 AM|
+-----------+-------+---------------+-----------------------------+----------------------+----------------------+
only showing top 20 rows
```

In [25]:
```
cols_to_show =

( new_fire_df
     .

)
```

```
+-----------+----------------------------+-------------------------------+-------------------+------------------
|Call Number|Call Type                   |Address                        |Received DtTm ts   |Response DtTm ts
+-----------+----------------------------+-------------------------------+-------------------+------------------
|220181779  |Other                       |17-TH DE HARO ST               |2022-01-18 13:38:10|2022-01-18 13:59:0
|211941035  |Citizen Assist / Service Call|0 Block of FRANKLIN ST         |2021-07-13 10:00:30|2021-07-13 10:05:4
|220180731  |Medical Incident            |0 Block of CARL ST             |2022-01-18 08:42:03|2022-01-18 08:47:1
|211940174  |Medical Incident            |500 Block of SOUTH VAN NESS AVE|2021-07-13 02:48:01|2021-07-13 02:54:1
|221200301  |Alarms                      |1000 Block of CHURCH ST        |2022-04-30 02:55:25|2022-04-30 03:01:1
|211230167  |Medical Incident            |HYDE ST/GROVE ST               |2021-05-03 02:55:59|2021-05-03 03:01:4
|220162991  |Medical Incident            |CHESTNUT ST/DIVISADERO ST      |2022-01-16 23:07:09|2022-01-16 23:14:1
|211212414  |Medical Incident            |1000 Block of VALENCIA ST      |2021-05-01 17:44:02|2021-05-01 18:06:5
|220161959  |Medical Incident            |0 Block of THOMAS MORE WAY     |2022-01-16 16:35:10|2022-01-16 16:44:0
|211210756  |Electrical Hazard           |400 Block of STOCKTON ST       |2021-05-01 09:03:25|2021-05-01 09:09:3
|210371768  |Medical Incident            |1900 Block of LEAVENWORTH ST   |2021-02-06 14:10:05|2021-02-06 14:15:3
|210360819  |Medical Incident            |CALL BOX: FS YB-BLDG 213       |2021-02-05 08:48:56|2021-02-05 08:55:1
|212892503  |Fuel Spill                  |15TH ST/GUERRERO ST            |2021-10-16 17:06:29|2021-10-16 17:11:3
|211900275  |Alarms                      |700 Block of 29TH AVE          |2021-07-09 03:47:03|2021-07-09 03:52:0
|211180439  |Alarms                      |2000 Block of MISSION ST       |2021-04-28 06:36:40|2021-04-28 06:41:5
|211870109  |Alarms                      |0 Block of 7TH ST              |2021-07-06 01:13:36|2021-07-06 01:19:0
|222023230  |High Angle Rescue           |0 Block of BOWL DR             |2022-07-21 22:16:08|2022-07-21 22:22:0
|211862605  |Medical Incident            |500 Block of VALENCIA ST       |2021-07-05 17:19:43|2021-07-05 17:26:2
|220112731  |Alarms                      |0 Block of OAKWOOD ST          |2022-01-11 18:12:20|2022-01-11 18:18:0
|221122156  |Elevator / Escalator Rescue |100 Block of TAYLOR ST         |2022-04-22 15:38:14|2022-04-22 15:43:3
+-----------+----------------------------+-------------------------------+-------------------+------------------
only showing top 20 rows
```

4) What were the most common call types, listed in descending order by count?

In [26]:
```
outcome_df = ( new_fire_df
               .

             )

outcome_df.show(truncate=False)
```

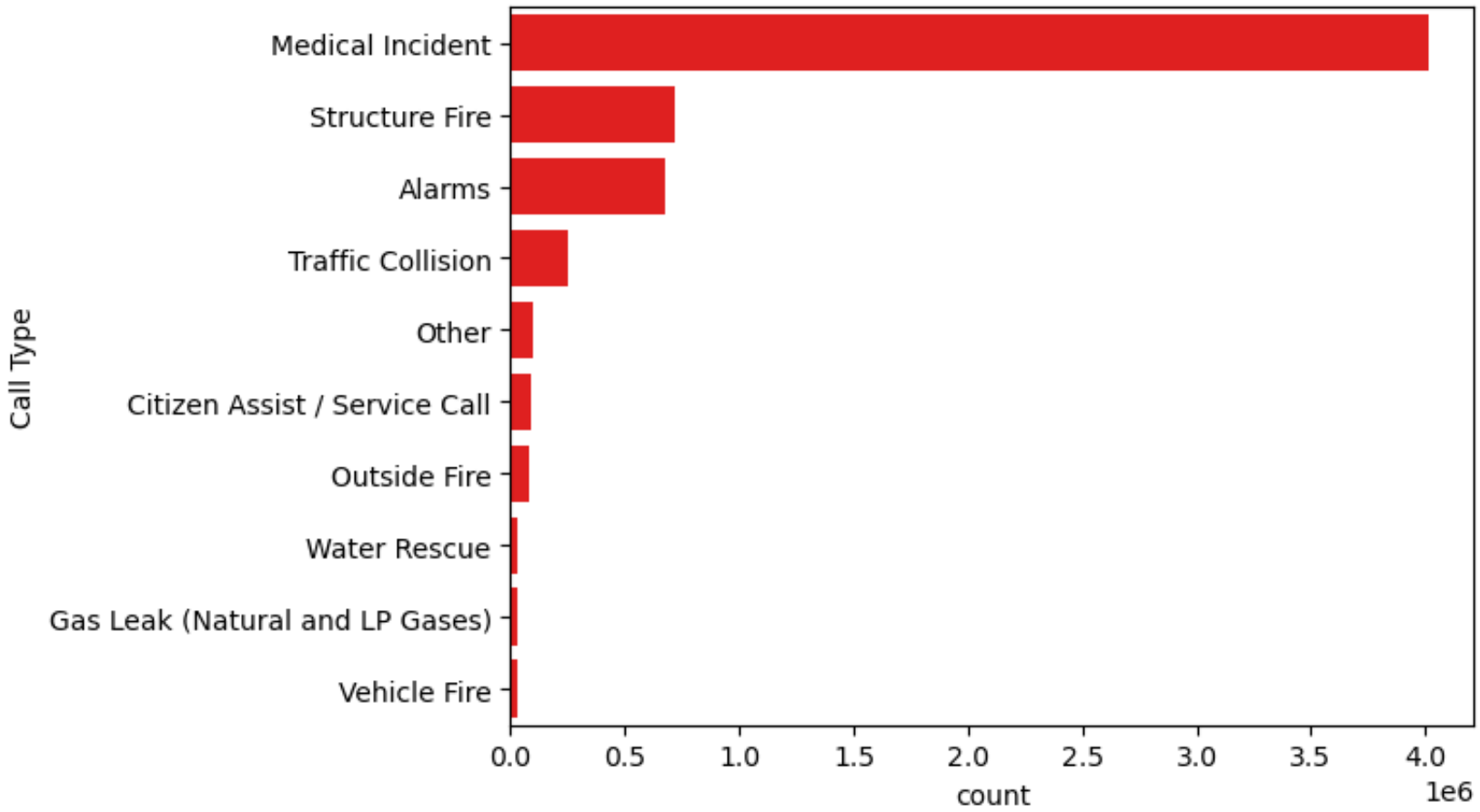
```
[Stage 18:=====================================================>   (17 + 1) / 18]
+---------------------------------+-------+
|Call Type                        |count  |
+---------------------------------+-------+
|Medical Incident                 |4012244|
|Structure Fire                   |715178 |
|Alarms                           |677178 |
|Traffic Collision                |247862 |
|Other                            |101940 |
|Citizen Assist / Service Call    |90988  |
|Outside Fire                     |79245  |
|Water Rescue                     |32353  |
|Gas Leak (Natural and LP Gases)  |27573  |
|Vehicle Fire                     |27435  |
|Electrical Hazard                |20112  |
|Elevator / Escalator Rescue      |16849  |
|Smoke Investigation (Outside)    |13867  |
|Odor (Strange / Unknown)         |13405  |
|Fuel Spill                       |6781   |
|HazMat                           |4313   |
|Industrial Accidents             |3275   |
|Explosion                        |2974   |
|Structure Fire / Smoke in Building|2180   |
|Train / Rail Incident            |1628   |
+---------------------------------+-------+
only showing top 20 rows
```

Visualizing the top 10 results

In [27]:
```
# PySpark DataFrame to pandas-on-Spark DataFrame and then to pandas,
# just for visualization

outcome_vis = outcome_df.limit(10).pandas_api().to_pandas()
```

In [28]:
```
plotHorizBar(df=outcome_vis, xcol="count", ycol="Call Type", colour="red")
```



5) What zip codes accounted for most common calls and what type were they?

1. Filter out by Call Type
2. Group them by Call Type and Zipcode of Incident
3. Count them and display in descending order

In [29]:
```
( new_fire_df
    .

)
```

```
[Stage 24:=====================================================>   (17 + 1) / 18]
+----------------+-------------------+------+
|Call Type       |Zipcode of Incident|count |
+----------------+-------------------+------+
|Medical Incident|94102              |582264|
|Medical Incident|94103              |538180|
|Medical Incident|94109              |347426|
|Medical Incident|94110              |346086|
|Medical Incident|94124              |206334|
|Medical Incident|94112              |194948|
|Medical Incident|94115              |166438|
|Medical Incident|94107              |147324|
|Medical Incident|94122              |144526|
|Medical Incident|94133              |135291|
|Medical Incident|94117              |124388|
|Medical Incident|94114              |116025|
|Medical Incident|94134              |114919|
|Medical Incident|94118              |109054|
|Medical Incident|94121              |102458|
|Medical Incident|94116              |89414 |
|Medical Incident|94132              |87845 |
|Medical Incident|94105              |81881 |
|Alarms          |94102              |77647 |
|Medical Incident|94108              |75248 |
+----------------+-------------------+------+
only showing top 20 rows
```

6) What neighbourhoods are in the two top zip codes from the listing in the previous question?

Probably these two zip codes are somehow related to contested neighbourhood with high reported crimes.

In [30]:
```
( new_fire_df
     .

)
```

```
[Stage 27:=====================================================>   (17 + 1) / 18]
+-----------------------------------+-------------------+
|Neighborhooods - Analysis Boundaries|Zipcode of Incident|
+-----------------------------------+-------------------+
|Financial District/South Beach     |94102              |
|South of Market                    |94102              |
|Mission Bay                        |94103              |
|Nob Hill                           |94102              |
|Castro/Upper Market                |94103              |
|Hayes Valley                       |94102              |
|Tenderloin                         |94103              |
|Western Addition                   |94102              |
|Potrero Hill                       |94103              |
|Mission                            |94103              |
|Hayes Valley                       |94103              |
|Tenderloin                         |94102              |
|South of Market                    |94103              |
|Financial District/South Beach     |94103              |
|Mission                            |94102              |
+-----------------------------------+-------------------+
```

Additional exercise

Using the given dataset, write down code to answer the following questions:

1. What was the sum of all calls, average, min and max of the response times for calls (from the moment call is received till response is acknowledged and unit is on route)?

   Hint: Use the functions sum(), avg(), min() and max()

2. How many distinct years of data is in the CSV file?

   Hint: Use the year() SQL Spark function off the timestamp of the Call Date column

3. What week of the year in 2017 had the most fire calls?

   Hint: Use the weekofyear() SQL Spark function off the timestamp of the Call Date column

Furthermore, create some visualizations to better understand the results obtained.

References

- Learning Spark - Lightning-Fast Data Analytics, 2nd Ed. J. Damji, B. Wenig, T. Das, and D. Lee. O'Reilly, 2020
- https://spark.apache.org/docs/latest
- https://docs.python.org/3/
- https://data.sfgov.org/Public-Safety/Fire-Department-Calls-for-Service/nuek-vuh3

In [ ]: