

**TECHNICAL DOCUMENT**

**CA2 – Programming for**

**Data Analysis**

**NAME: Ankit Butola**

**STUDENT ID: 10590492**

***Pandas library in Python***

Table of Contents

[**1.** **Scope** 2](#_Toc90134420)

[1.1 Vision and Goals 2](#_Toc90134421)

[1.2 Stakeholders 2](#_Toc90134422)

[1.3 Business Requirements 3](#_Toc90134423)

[**2.** **Technical Design** 3](#_Toc90134424)

[2.1 ETL Architecture 3](#_Toc90134425)

[2.2 DETAILED PANDAS OPERATION 6](#_Toc90134426)

[**3.** **DATA MODEL** 7](#_Toc90134427)

[**4.** **Testing** 7](#_Toc90134428)

[**REMOVING “,” with an EMPTY STRING** 7](#_Toc90134429)

[**Removing “-“ with an Empty string** 8](#_Toc90134430)

[**Drop rows for the columns (AH, MAV, CD) where at least one row value is NULL.** 9](#_Toc90134431)

[**Drop any duplicate rows (except for the first occurrence)** 10](#_Toc90134432)

[**5.** **Reflection on Learnings** 11](#_Toc90134433)

[**6.** **References** 11](#_Toc90134434)

# **Scope**

## 1.1 Vision and Goals

The primary goal of this project is to clean the data in preparation for further processing and to generate business insights based on meaningful and complete data.

## 1.2 Stakeholders

1. Business Analyst

2. System Owner

3. System Custodian

4. Business Analyst

5. Subject Matter Expert

## 1.3 Business Requirements

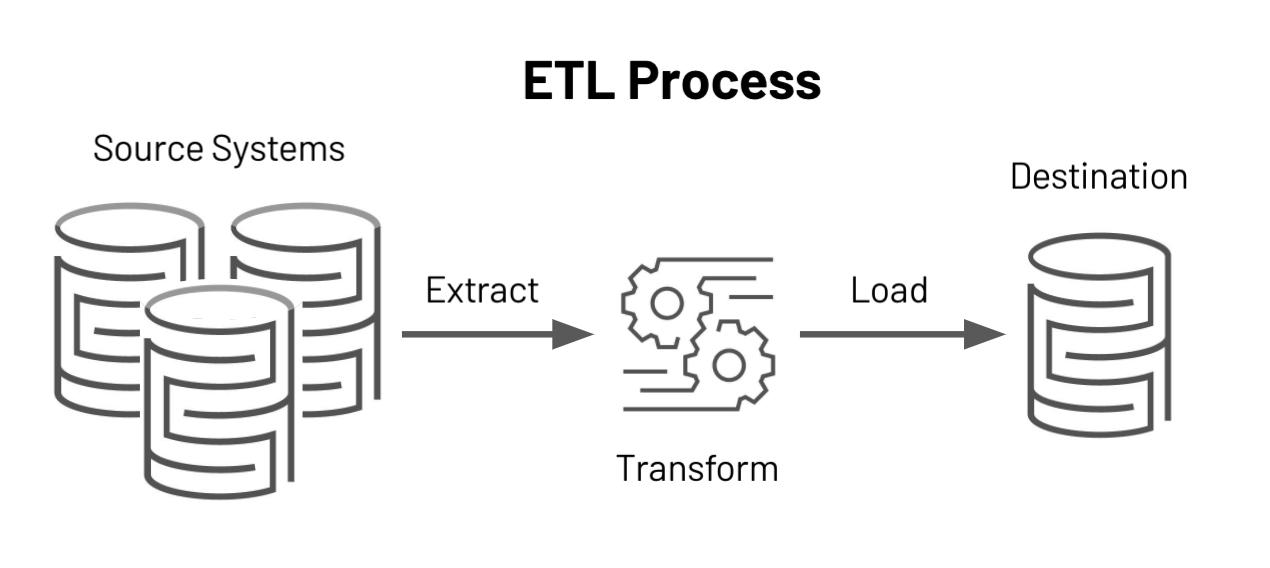
Cleaning data and importing it into a database in SQL Server in order to

generate helpful insights regarding the government's emergency response

times.

# **Technical Design**

## 2.1 ETL Architecture



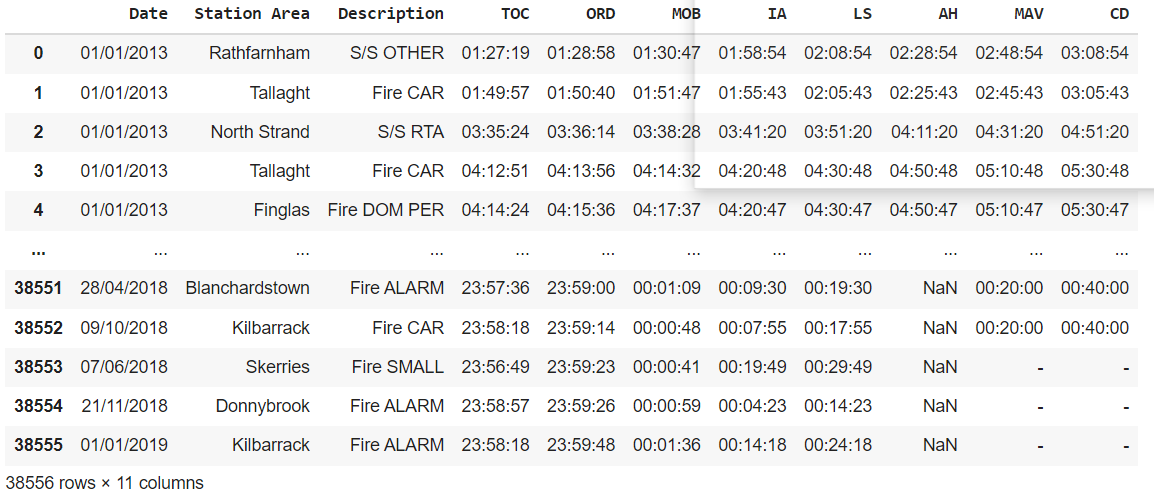
**Extraction:**

The process of extractng the data from a Source

file(“FireBrigadeAndAmbulanceCallOuts.csv”).

Given Source is a csv file. Data being extracted using method **df.read\_csv** in Pandas.

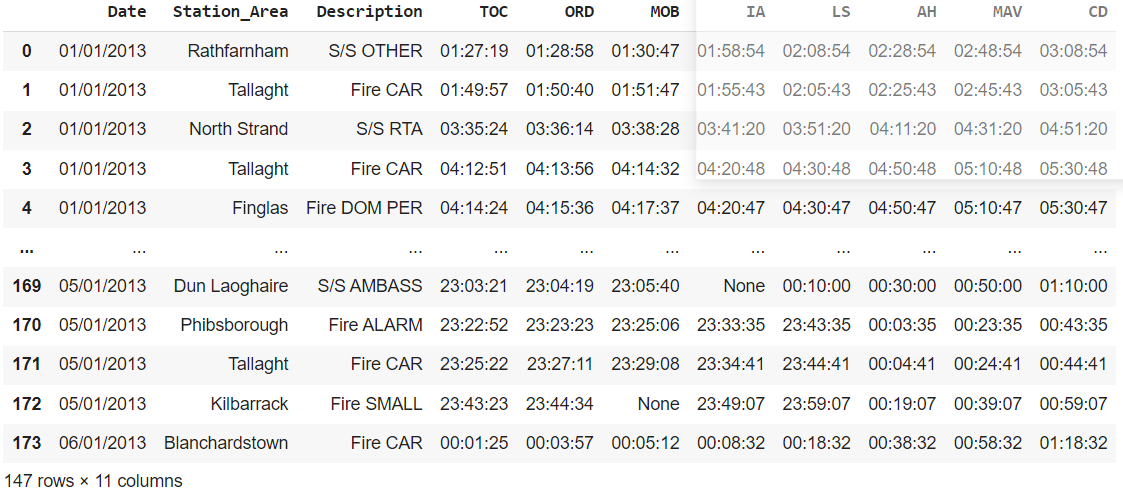
**Extracted Data,**

****

**Transformation:**

Extractd data being transformed into cleaner data by removing special characters and missing values from the dataset, using Pandas Python Library.

Data after Transformation,



**Load:**

Transformed data has been loaded into SQL server by making connection string using pypyodbc driver in pandas.

Connection string :

connection = pypyodbc.connect('Driver={ODBC Driver 17 for SQL Server};'

'Server=20.71.187.229;'

'Database=master;'

'encrypt=yes;'

'TrustServerCertificate=yes;'

'UID=SA;'

'PWD=Today@911',autocommit = True)

cursor = connection.cursor()

SQLCommand = ("""CREATE TABLE emergencyCallOutRecords (Date varchar (256),

Station\_Area varchar (256),

Description varchar (256),

TOC varchar (256),

ORD varchar (256) ,

MOB varchar (256),

IA varchar (256),

LS varchar (256),

AH varchar (256),

MAV varchar (256),

CD varchar (256))

""")

cursor.execute(SQLCommand)

print(Table created successfully)

for index, row in df.iterrows():

query = "INSERT INTO emergencyCallOutRecords (Date, Station\_Area, Description, TOC, ORD, MOB, IA, LS, AH, MAV, CD) VALUES(?,?,?,?,?,?,?,?,?,?,?)"

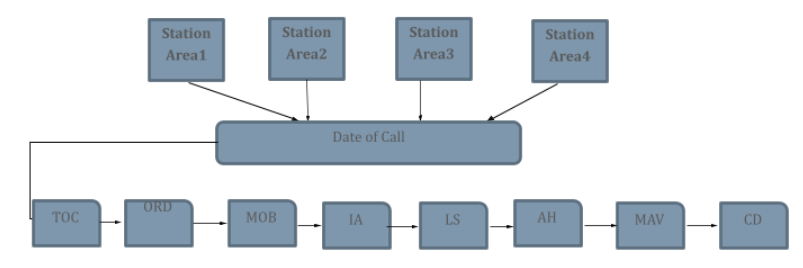
cursor.execute(query, (row.Date, row.Station\_Area, row. Description, row.TOC, row.ORD, row.MOB, row.IA, row.LS, row.AH, row.MAV, row.CD))

connection.close()

## 2.2 DETAILED PANDAS OPERATION

* Replacing “,” with empty strings with the use of replace function
* Replacing “-“ with empty strings with the use of replace function.
* Removing rows with Null values in columns anyone of the columns AH, MAV, CD by using dropna function with parameters as {how = ‘any’,subset=[column\_names]}
* Removing duplicity of rows using drop\_duplicates function with a parameter as keep=’first

# **DATA MODEL**



TOC : The time the call is received in the control center

ORD : The time the vehicle is ordered, i.e., mobilized to the incident by a control operator.

MOB : The time at which the vehicle is mobile to incident (the vehicle has started to move)

IA : The time the vehicle is in attendance (the vehicle is stopped at the incident)

LS : The time the ambulance is leaving scene for hospital

AH : The time at hospital (ambulance has arrived at hospital)

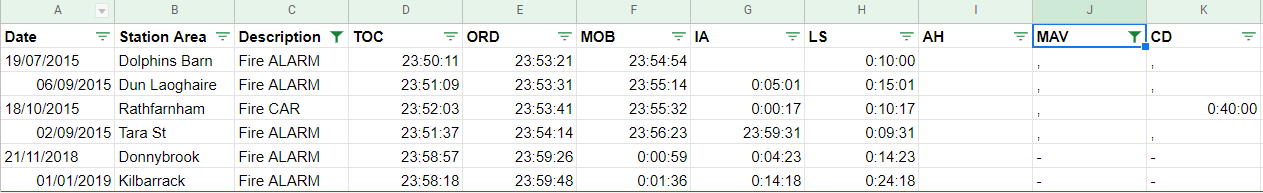
MAV : The time at which the vehicle is mobile and available (vehicle heading back to station)

CD : The time at which the vehicle is closing down (back at station, vehicle radio is being shut down)

# **Testing**

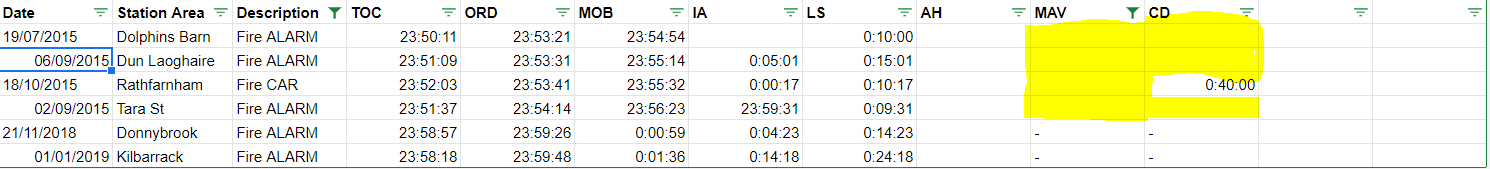
## **REMOVING “,” with an EMPTY STRING**

Without Replacement:



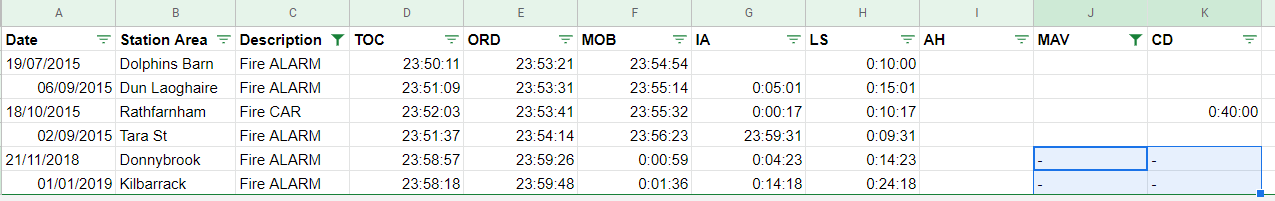
With replacement:

Code: df = df.replace(',',"")



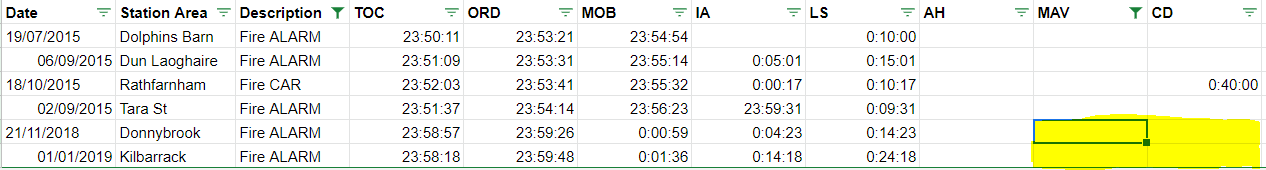
## **Removing “-“ with an Empty string**

Without Replacement:



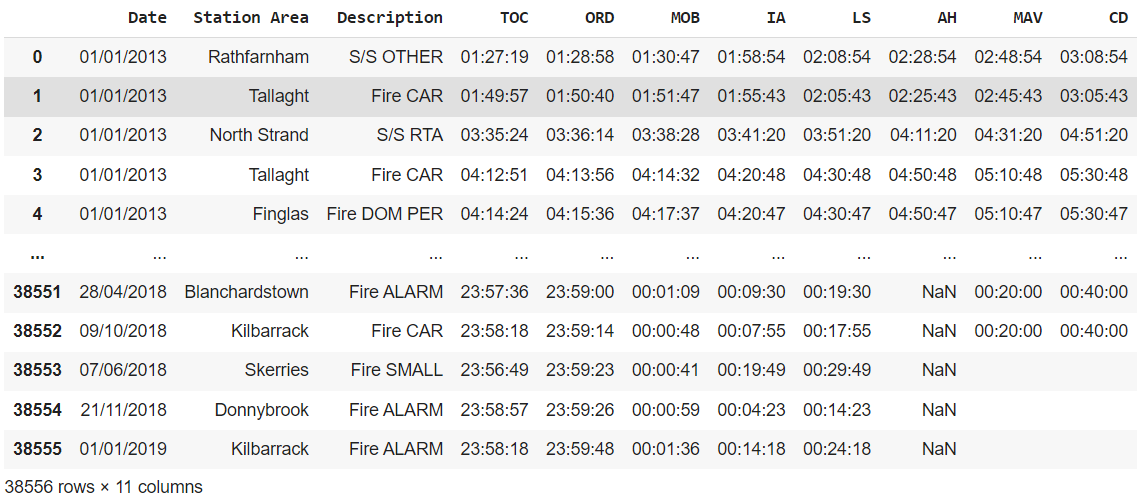
With replacement:

Code: df = df.replace('-',"")

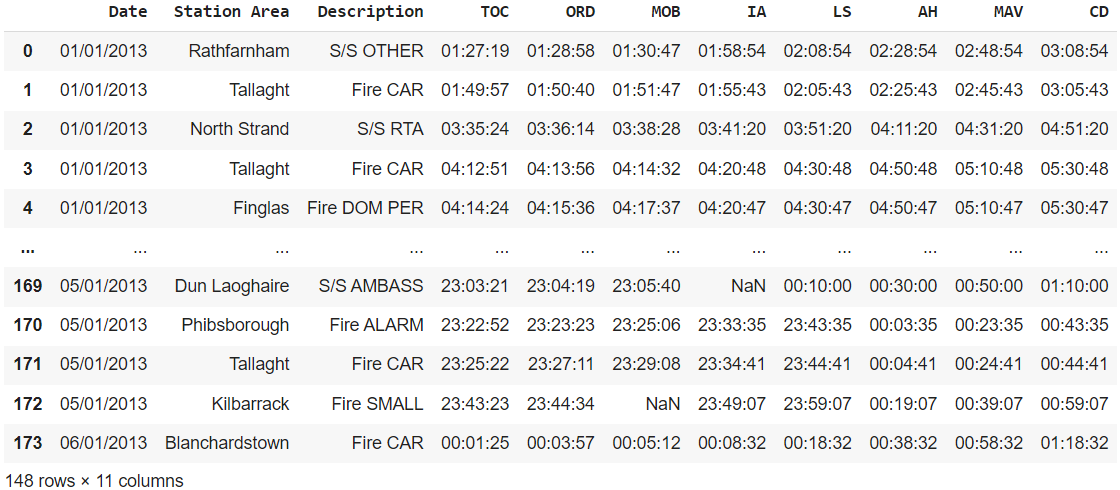


## **Drop rows for the columns (AH, MAV, CD) where at least one row value is NULL.**

Without Dropping Null values in Columns AH, MAV, CD :

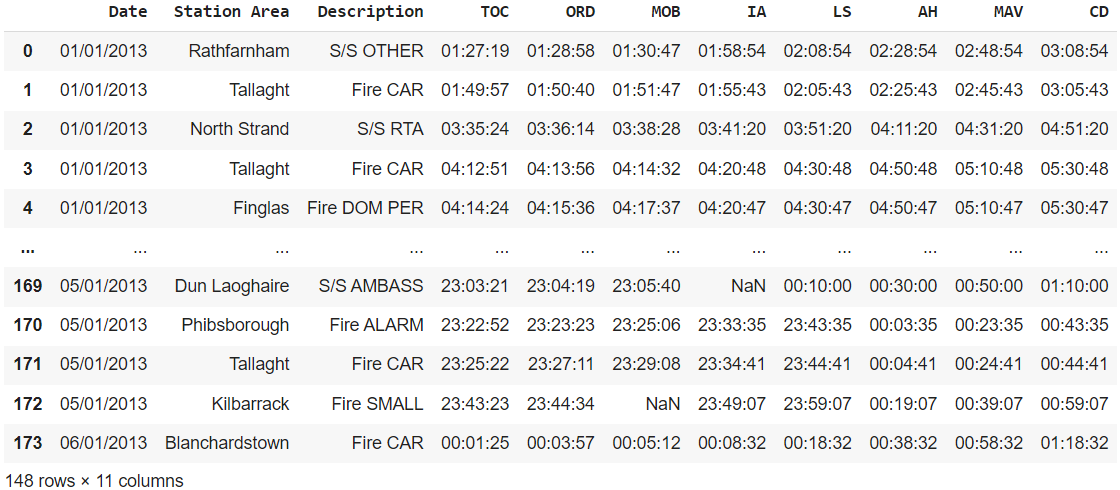


After Dropping Null values :



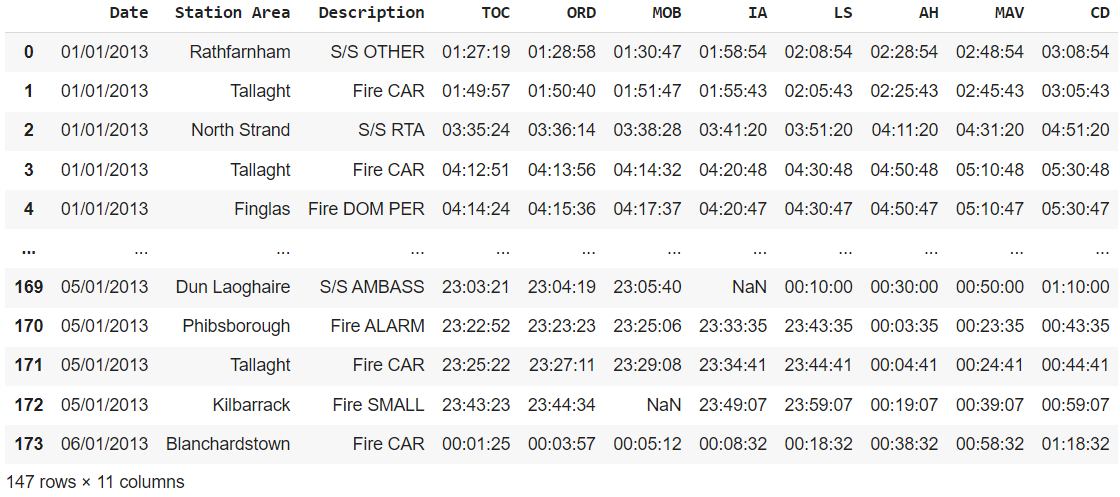
## **Drop any duplicate rows (except for the first occurrence)**

Without dropping duplicate rows:



After removing duplicate rows:

Code: df.drop\_duplicates(keep = 'first' ,inplace = True)



# **Reflection on Learnings**

1) Understanding of ETL and its applications

2) Data cleaning techniques using Pandas as library in Python

3) Generating useful insights from a data at our disposal

4) Loading data into a table in sql server using a connection string.

5) Data Extraction and Transformation techniques/

# **References**

<https://www.analyticsvidhya.com/blog/2016/01/12-pandas-techniques-python-data-manipulation/>

<https://realpython.com/python-pandas-tricks/>