**Data Cleaning: Colorado Police logs**

**Metadata**

Column Name: Definition

**Incident\_ID:** Unique identifier for call.

**ResponseDate**: Date and time that the call was received.

**CaseNumber**: Case number associated with the incident. Case numbers can be added up to 24 hours after a call's response date. Blank cells indicate there was no police report generated for the particular call.

**HundredBlock**: Address (rounded up to hundred block location) of the incident.

1. The data is loaded in the csv file format into python

**data=pd.read\_csv('filename with correct extension')**

2. The shape(dimensions) of the data file are found out

**data.shape**

3. The data type of the columns are then found

**data.dtypes**

4. Head and tail of data

**data.head()**

**data.tail()**

5. Dropping columns in the beginning or the end

**data.drop(data.tail().index,inplace=True)**

**data.drop(data.head().index,inplace=True)**

6. Some values in the incident id and response date columns are interchanged. These need to be corrected

From the value count of unique response dates, we find that the incident id’s 1,2,3,4 are the values which have been interchanged with the dates.

**data[['Incident\_ID','ResponseDate']] = data[['ResponseDate','Incident\_ID']].where(data['ResponseDate'] == '1', data[['Incident\_ID','ResponseDate']].values)**

**data[['Incident\_ID','ResponseDate']] = data[['ResponseDate','Incident\_ID']].where(data['ResponseDate'] == '2', data[['Incident\_ID','ResponseDate']].values)**

**data[['Incident\_ID','ResponseDate']] = data[['ResponseDate','Incident\_ID']].where(data['ResponseDate'] == '3', data[['Incident\_ID','ResponseDate']].values)**

**data[['Incident\_ID','ResponseDate']] = data[['ResponseDate','Incident\_ID']].where(data['ResponseDate'] == '4', data[['Incident\_ID','ResponseDate']].values)**

7. Some values in case number and problems are interchanged. Case number is in the format of yyyymmdd-number

So, all the case numbers begin with ‘2’

**data[['CaseNumber','Problem']] = data[['Problem','CaseNumber']].where(data['Problem'].astype(str).str[0] == '2', data[['CaseNumber','Problem']].values)**

8. There are problems with Case numbers, some of the null values in problems are interchanged with case number and also the case number empty values are replaced with NaN

**data[['CaseNumber','Problem']] = data[['Problem','CaseNumber']].where(data['Problem'].isnull(), data[['CaseNumber','Problem']].values)**

**data['CaseNumber']=data['CaseNumber'].fillna('NaN')**

9. Splitting Problem into code and the problem

**new\_data=data['Problem'].str.split("-", n = 1, expand = True)**

**data["Code"]= new\_data[0]**

**data["Problem"]= new\_data[1]**

10. Defining a function to get the status of the case based off on the last three characters of the problem column

**def status(data):**

**if 'I/P'in data:**

**return (data[-3:])**

**elif 'J/O' in data:**

**return (data[-3:])**

**else:**

**return "Completed"**

**data['Status']=data['Problem'].apply(status)**

11. Removing the status of the problem from the problem column

**def problem(data):**

**if 'I/P'in data:**

**return (data[:-3])**

**elif 'J/O' in data:**

**return (data[:-3])**

**else:**

**return (data)**

**data['Problem']=data['Problem'].apply(problem)**

12. Some of the columns have a C next to them that needs to be removed

**def elimC(data):**

**if 'C'in data[-1:]:**

**return (data[:-1])**

**else:**

**return (data)**

**data['Problem']=data['Problem'].apply(elimC)**

13. ResponseDate column is a string, wwe convert it to datetime and then the datetime is converted back to a string with just the date part.

**import datetime as dt**

**data['datetime'] = pd.to\_datetime(data['ResponseDate'])**

**data['datetime'] = data['datetime'].dt.strftime("%m/%d/%Y")**

14. Converting time to a 24 hr format

**data['Time'] = pd.to\_datetime(data['ResponseDate'])**

**data['Time'] = data['Time'].dt.strftime("%:%M:%S")**