PE29 Vasu Kalariya

IMLA Lab Assi 2 Data Preprocessing

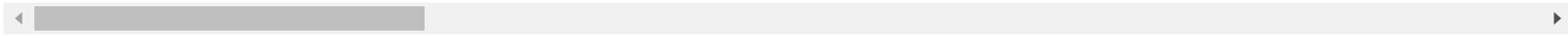
In [46]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [47]:
```
ds = pd.read_csv('FireData.csv')
ds
```

Out[47]:

	_id	Area_of_Origin	Building_Status	Business_Impact	Civilian_Casualties	Count_of_Persons_Rescued	Estimated_Dollar_Loss
0	190306	81 - Engine Area	NaN	NaN	0	0	1000.0
1	190307	22 - Sleeping Area or Bedroom (inc. patients r...	01 - Normal (no change)	8 - Not applicable (not a business)	0	0	20.0
2	190308	81 - Engine Area	NaN	NaN	0	0	5000.0
3	190309	99 - Undetermined (formerly 98)	NaN	NaN	0	0	7500.0
4	190310	81 - Engine Area	NaN	NaN	0	0	10000.0
...	...	...	...	...	...	...	...
12682	202988	81 - Engine Area	NaN	NaN	0	0	500.0
12683	202989	81 - Engine Area	NaN	NaN	0	0	40000.0
12684	202990	27 - Laundry Area	01 - Normal (no change)	1 - No business interruption	0	0	2000.0
12685	202991	21 - Living Area (e.g. living, TV, recreation,...	01 - Normal (no change)	8 - Not applicable (not a business)	1	0	25.0
12686	202992	97 - Other - unclassified	08 - Not Applicable	8 - Not applicable (not a business)	1	0	10.0

12687 rows × 43 columns



In [48]:
```
ds.isnull().sum()
```

Out[48]:

_id	0
Area_of_Origin	0
Building_Status	3482
Business_Impact	3484
Civilian_Casualties	0
Count_of_Persons_Rescued	0
Estimated_Dollar_Loss	1
Estimated_Number_Of_Persons_Displaced	3483
Exposures	12408
Ext_agent_app_or_defer_time	0
Extent_Of_Fire	3484
Final_Incident_Type	0
Fire_Alarm_System_Impact_on_Evacuation	3484
Fire_Alarm_System_Operation	3484
Fire_Alarm_System_Presence	3484
Fire_Under_Control_Time	1
Ignition_Source	0
Incident_Number	0
Incident_Station_Area	0
Incident_Ward	74
Initial_CAD_Event_Type	0
Intersection	1
Last_TFS_Unit_Clear_Time	0
Latitude	1
Level_Of_Origin	3484
Longitude	1
Material_First_Ignited	0
Method_Of_Fire_Control	0
Number_of_responding_apparatus	0
Number_of_responding_personnel	0
Possible_Cause	0
Property_Use	1
Smoke_Alarm_at_Fire_Origin	3484
Smoke_Alarm_at_Fire_Origin_Alarm_Failure	3484
Smoke_Alarm_at_Fire_Origin_Alarm_Type	3484
Smoke_Alarm_Impact_on_Persons_Evacuating_Impact_on_Evacuation	3484
Smoke_Spread	3484
Sprinkler_System_Operation	3484
Sprinkler_System_Presence	3484
Status_of_Fire_On_Arrival	0
TFS_Alarm_Time	0

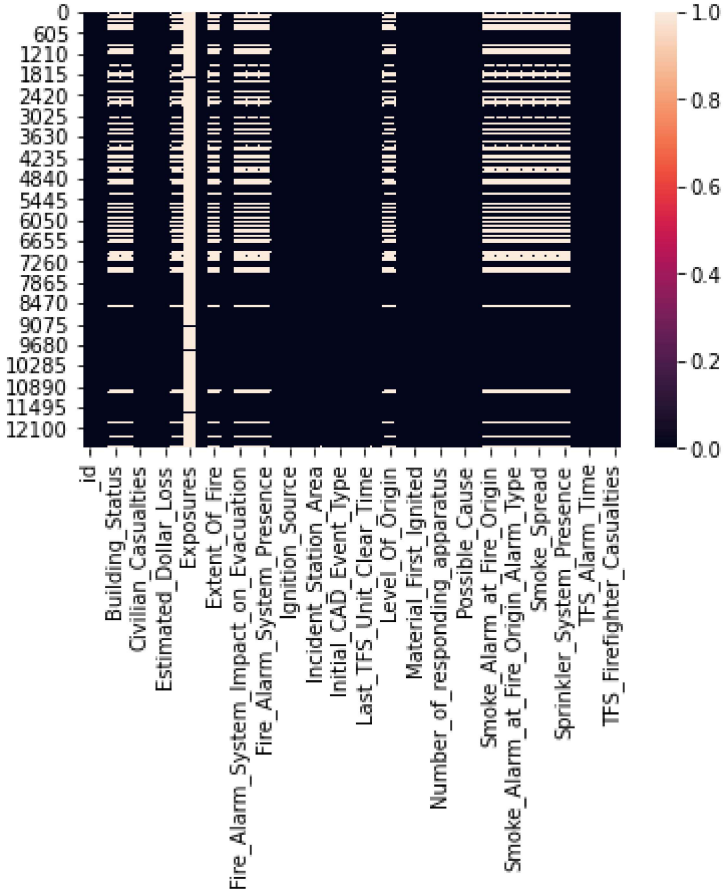
```
TFS_Arrival_Time                                                    0
TFS_Firefighter_Casualties                                          0
dtype: int64
```

In [49]:
```
ds.shape
```

Out[49]:
```
(12687, 43)
```

In [50]:
```
sns.heatmap(ds.isnull())
```

Out[50]:
```
<AxesSubplot:>
```



In [51]:
```
ds.dropna(subset=['Building_Status'], inplace = True)
```

In [52]:
```
ds.isnull().sum()
```

Out[52]:

_id	0
Area_of_Origin	0
Building_Status	0
Business_Impact	2
Civilian_Casualties	0
Count_of_Persons_Rescued	0
Estimated_Dollar_Loss	0
Estimated_Number_Of_Persons_Displaced	1
Exposures	8994
Ext_agent_app_or_defer_time	0
Extent_Of_Fire	2
Final_Incident_Type	0
Fire_Alarm_System_Impact_on_Evacuation	2
Fire_Alarm_System_Operation	2
Fire_Alarm_System_Presence	2
Fire_Under_Control_Time	0
Ignition_Source	0
Incident_Number	0
Incident_Station_Area	0
Incident_Ward	30
Initial_CAD_Event_Type	0
Intersection	0
Last_TFS_Unit_Clear_Time	0
Latitude	0
Level_Of_Origin	2
Longitude	0
Material_First_Ignited	0
Method_Of_Fire_Control	0
Number_of_responding_apparatus	0
Number_of_responding_personnel	0
Possible_Cause	0
Property_Use	0
Smoke_Alarm_at_Fire_Origin	2
Smoke_Alarm_at_Fire_Origin_Alarm_Failure	2
Smoke_Alarm_at_Fire_Origin_Alarm_Type	2
Smoke_Alarm_Impact_on_Persons_Evacuating_Impact_on_Evacuation	2
Smoke_Spread	2
Sprinkler_System_Operation	2
Sprinkler_System_Presence	2
Status_of_Fire_On_Arrival	0

```
TFS_Alarm_Time                                  0
TFS_Arrival_Time                                0
TFS_Firefighter_Casualties                      0
dtype: int64
```

In [53]:
```
ds['Exposures'].unique()
```

Out[53]:
```
array([nan,  1.,  3.,  2.,  4.,  6.,  5.,  7.])
```

In [54]:
```
Exposures1 = ds.iloc[:,8:9].values
Exposures2 = ds.iloc[:,8:9].values
Exposures3 = ds.iloc[:,8:9].values
Exposures1
```

Out[54]:
```
array([[nan],
       [nan],
       [nan],
       ...,
       [nan],
       [nan],
       [nan]])
```

Imputer for Missing values

In [55]:
```
from sklearn.impute import SimpleImputer
si1 = SimpleImputer(missing_values=np.nan,strategy='mean')
si1.fit(Exposures1)
Exposures1 = si1.transform(Exposures1)
Exposures1 = pd.DataFrame(Exposures1, columns=['Exposures1'])
Exposures1
```

Out[55]:

	Exposures1
0	1.649289
1	1.649289
2	1.649289
3	1.649289
4	1.649289
...	...
9200	1.649289
9201	1.649289
9202	1.649289
9203	1.649289
9204	1.649289

9205 rows × 1 columns

In [56]:
```
si2 = SimpleImputer(missing_values=np.nan,strategy='median')
si2.fit(Exposures2)
Exposures2 = si2.transform(Exposures2)
Exposures2 = pd.DataFrame(Exposures2, columns=['Exposures2'])
Exposures2
```

Out[56]:

	Exposures2
0	1.0
1	1.0
2	1.0
3	1.0
4	1.0
...	...
9200	1.0
9201	1.0
9202	1.0
9203	1.0
9204	1.0

9205 rows × 1 columns

In [57]:
```
ds.isnull().sum()
```

Out[57]:

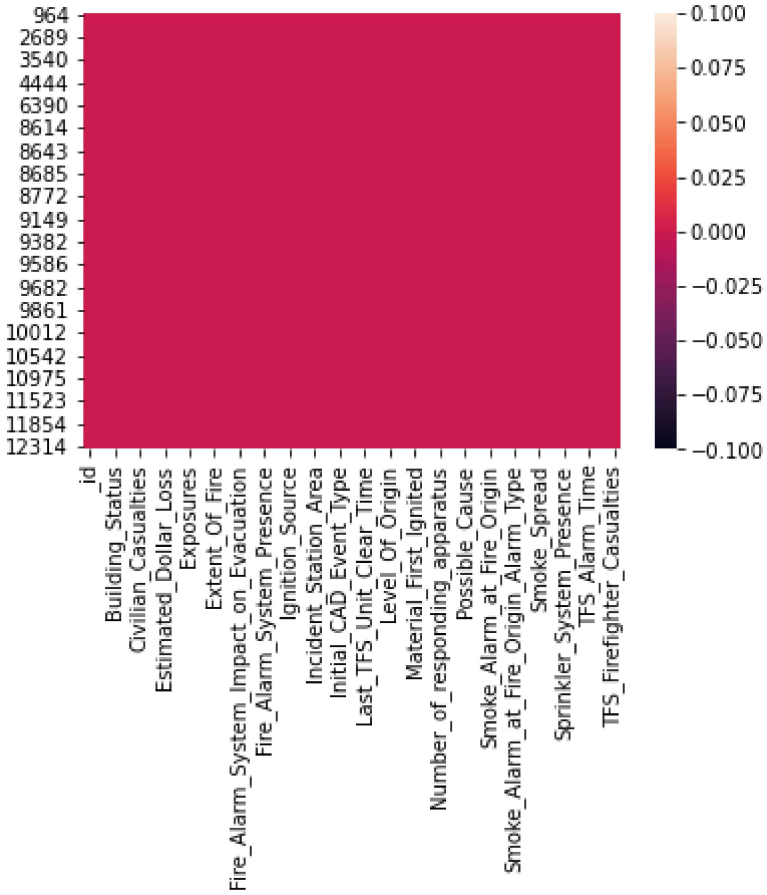
_id	0
Area_of_Origin	0
Building_Status	0
Business_Impact	2
Civilian_Casualties	0
Count_of_Persons_Rescued	0
Estimated_Dollar_Loss	0
Estimated_Number_Of_Persons_Displaced	1
Exposures	8994
Ext_agent_app_or_defer_time	0
Extent_Of_Fire	2
Final_Incident_Type	0
Fire_Alarm_System_Impact_on_Evacuation	2
Fire_Alarm_System_Operation	2
Fire_Alarm_System_Presence	2
Fire_Under_Control_Time	0
Ignition_Source	0
Incident_Number	0
Incident_Station_Area	0
Incident_Ward	30
Initial_CAD_Event_Type	0
Intersection	0
Last_TFS_Unit_Clear_Time	0
Latitude	0
Level_Of_Origin	2
Longitude	0
Material_First_Ignited	0
Method_Of_Fire_Control	0
Number_of_responding_apparatus	0
Number_of_responding_personnel	0
Possible_Cause	0
Property_Use	0
Smoke_Alarm_at_Fire_Origin	2
Smoke_Alarm_at_Fire_Origin_Alarm_Failure	2
Smoke_Alarm_at_Fire_Origin_Alarm_Type	2
Smoke_Alarm_Impact_on_Persons_Evacuating_Impact_on_Evacuation	2
Smoke_Spread	2
Sprinkler_System_Operation	2
Sprinkler_System_Presence	2
Status_of_Fire_On_Arrival	0
TFS_Alarm_Time	0

```
TFS_Arrival_Time                                                    0
TFS_Firefighter_Casualties                                          0
dtype: int64
```

In [58]:
```
ds.dropna(axis=0, how="any", thresh=None, subset=None, inplace=True)
```

In [59]:
```
sns.heatmap(ds.isnull())
```

Out[59]: <AxesSubplot:>



In [60]:
```
ds = ds.reset_index()
```

In [61]:
```
ds['Estimated_Dollar_Loss'] = ds['Estimated_Dollar_Loss'].astype('int64')
ds['Estimated_Dollar_Loss'].unique()
```

Out[61]:
```
array([   2000,     200,    5000,   70000,  100000,    7500,   10000,
            400,   90000,    1000,   20000,    2500,   25000,   50000,
          30000,   40000,  350000,     500,   60000,    9000, 1000000,
          80000,  500000,  400000, 5000000,  250000,  200000,  750000,
         800000,  125000,  700000,   34000,  300000,    9999, 2000000,
          75000, 1500000,  150000,    4000,   15000,    7000,     250,
         120000,      20,  175000,   18000,   65000,   12000,   35000,
          49800,  600000,   55000,     100, 1850000,    8000], dtype=int64)
```

Normalisation using Max Min, L1 , L2 , Zscore

In [62]:
```
Estimated_Dollar_Loss = ds.iloc[:, 6:7].values

from sklearn import preprocessing
min_max = preprocessing.MinMaxScaler()
Estimated_Dollar_Loss= min_max.fit_transform(Estimated_Dollar_Loss)
Estimated_Dollar_Loss = pd.DataFrame(Estimated_Dollar_Loss,columns = ['Estimated_Dollar_Loss'])
Estimated_Dollar_Loss
```

Out[62]:

	Estimated_Dollar_Loss
0	0.0
1	0.0
2	0.0
3	0.0
4	0.0
...	...
206	0.0
207	0.0
208	0.0
209	0.0
210	0.0

211 rows × 1 columns

In [63]:
```
Estimated_Dollar_Loss2 = ds.iloc[:, 6:7].values

from sklearn.preprocessing import Normalizer
Data_normalizer= Normalizer(norm='l1').fit(Estimated_Dollar_Loss2)
Estimated_Dollar_Loss2 = Data_normalizer.transform(Estimated_Dollar_Loss2)
Estimated_Dollar_Loss2 = pd.DataFrame(Estimated_Dollar_Loss2,columns = ['Estimated_Dollar_Loss2'])
Estimated_Dollar_Loss2
```

Out[63]:

	Estimated_Dollar_Loss2
0	0.0
1	0.0
2	0.0
3	0.0
4	0.0
...	...
206	0.0
207	0.0
208	0.0
209	0.0
210	0.0

211 rows × 1 columns

In [64]:
```
Estimated_Dollar_Loss3 = ds.iloc[:, 6:7].values

from sklearn.preprocessing import Normalizer
Data_normalizer= Normalizer(norm='l2').fit(Estimated_Dollar_Loss3)
Estimated_Dollar_Loss3 = Data_normalizer.transform(Estimated_Dollar_Loss3)
Estimated_Dollar_Loss3 = pd.DataFrame(Estimated_Dollar_Loss3,columns = ['Estimated_Dollar_Loss3'])
Estimated_Dollar_Loss3
```

Out[64]:

	Estimated_Dollar_Loss3
0	0.0
1	0.0
2	0.0
3	0.0
4	0.0
...	...
206	0.0
207	0.0
208	0.0
209	0.0
210	0.0

211 rows × 1 columns

In [65]:
```
Estimated_Number_Of_Persons_Displaced = ds.iloc[:, 7:8].values

from scipy import stats
Estimated_Number_Of_Persons_Displaced = stats.zscore(Estimated_Number_Of_Persons_Displaced)
Estimated_Number_Of_Persons_Displaced = pd.DataFrame(Estimated_Number_Of_Persons_Displaced,columns = ['Estimated
Estimated_Number_Of_Persons_Displaced
```

Out[65]:

	Estimated_Number_Of_Persons_Displaced
0	-0.467214
1	-0.471048
2	-0.460825
3	-0.322387
4	-0.258493
...	...
206	-0.301089
207	-0.466149
208	-0.454435
209	-0.450176
210	-0.450176

211 rows × 1 columns

Feature Selection using Correlation Coefficient

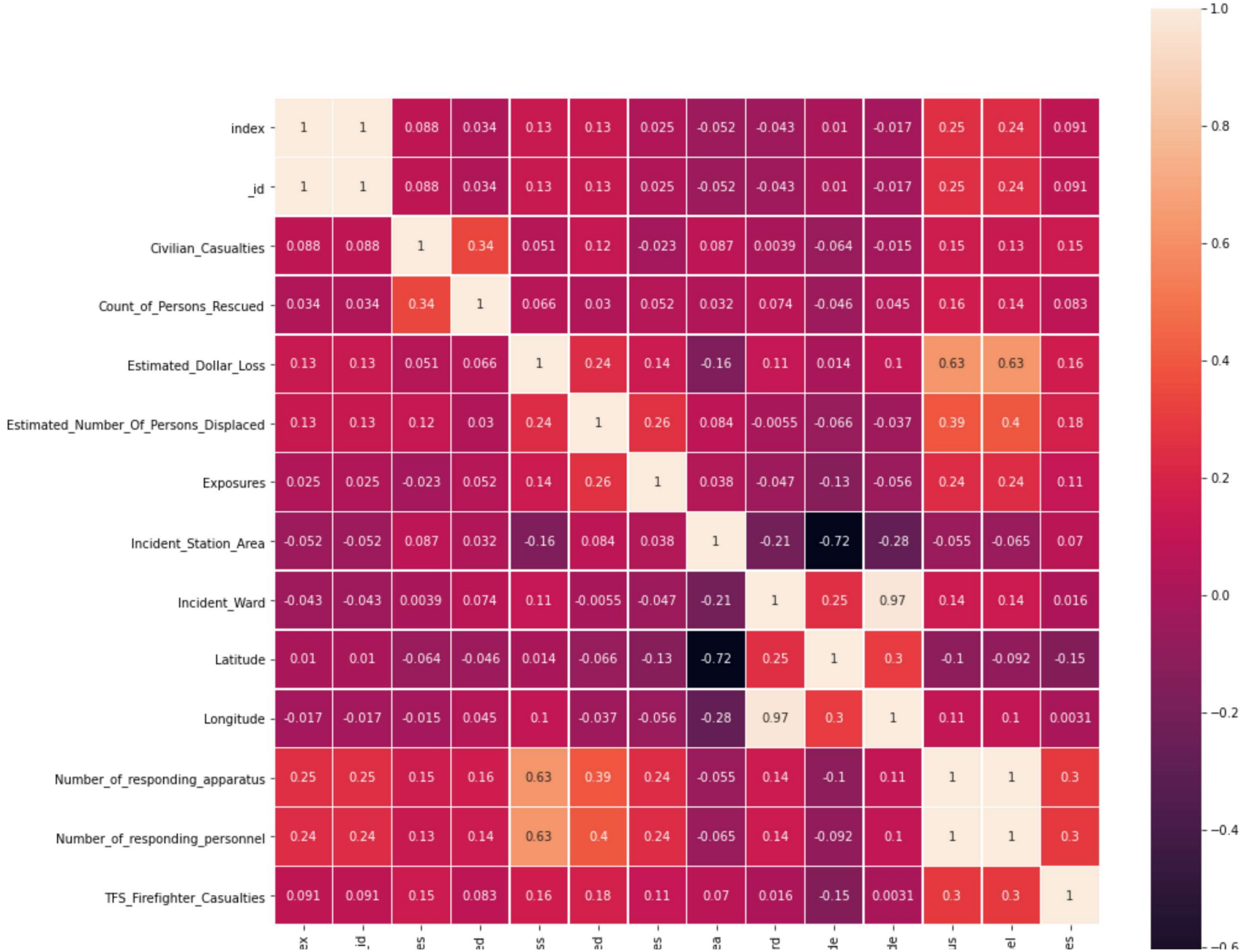
In [66]:
```
ds.corr(method ='pearson')
```

Out[66]:

	index	_id	Civilian_Casualties	Count_of_Persons_Rescued	Estimated_Dollar_Loss	Est
index	1.000000	1.000000	0.088213	0.034237	0.128786	
_id	1.000000	1.000000	0.088213	0.034237	0.128786	
Civilian_Casualties	0.088213	0.088213	1.000000	0.336775	0.050592	
Count_of_Persons_Rescued	0.034237	0.034237	0.336775	1.000000	0.066351	
Estimated_Dollar_Loss	0.128786	0.128786	0.050592	0.066351	1.000000	
Estimated_Number_Of_Persons_Displaced	0.125300	0.125300	0.120813	0.030255	0.236972	
Exposures	0.025089	0.025089	-0.022841	0.051705	0.135748	
Incident_Station_Area	-0.051774	-0.051774	0.087232	0.032444	-0.160756	
Incident_Ward	-0.043413	-0.043413	0.003947	0.074149	0.109913	
Latitude	0.010110	0.010110	-0.064315	-0.046431	0.013641	
Longitude	-0.017260	-0.017260	-0.014582	0.044854	0.103475	
Number_of_responding_apparatus	0.249495	0.249495	0.152842	0.163182	0.629676	
Number_of_responding_personnel	0.239354	0.239354	0.134114	0.144415	0.630974	
TFS_Firefighter_Casualties	0.091333	0.091333	0.147323	0.083274	0.162295	

In [67]:
```
plt.subplots(figsize=(15,15))
sns.heatmap(ds.corr(), annot = True,annot_kws={'size': 10},linewidths=.5,square=True)
```

Out[67]: <AxesSubplot:>



Encoding of categorical values

In [68]:
```
Final_Incident_Type = ds.loc[:, ['Final_Incident_Type']].values

from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
Final_Incident_Type = le.fit_transform(Final_Incident_Type)
Final_Incident_Type = pd.DataFrame(Final_Incident_Type,columns = ['Final_Incident_Type(Explosion)'])
Final_Incident_Type
```

```
C:\Users\kalar\anaconda3\lib\site-packages\sklearn\utils\validation.py:72: DataConversionWarning: A column-vec
tor y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example usin
g ravel().
  return f(**kwargs)
```

Out[68]:

	Final_Incident_Type(Explosion)
0	0
1	0
2	0
3	0
4	0
...	...
206	0
207	0
208	0
209	0
210	0

211 rows × 1 columns

In [69]:
```
Method_Of_Fire_Control = pd.get_dummies(ds['Method_Of_Fire_Control'])
Method_Of_Fire_Control
```

Out[69]:

	1 - Extinguished by fire department	3 - Extinguished by occupant	4 - Fire self extinguished	5 - Action taken unclassified
0	1	0	0	0
1	0	0	1	0
2	1	0	0	0
3	1	0	0	0
4	1	0	0	0
...	...	...	...	...
206	1	0	0	0
207	1	0	0	0
208	1	0	0	0
209	1	0	0	0
210	1	0	0	0

211 rows × 4 columns

In [70]:
```
Method_Of_Fire_Control1 = ds.loc[:,['Method_Of_Fire_Control']].values

from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder

ct = ColumnTransformer(transformers = [('encoder',OneHotEncoder(), [0])],remainder='passthrough')
Method_Of_Fire_Control1 = np.array(ct.fit_transform(Method_Of_Fire_Control1))
print(Method_Of_Fire_Control1)
```

```
  (0, 0)        1.0
  (1, 2)        1.0
  (2, 0)        1.0
  (3, 0)        1.0
  (4, 0)        1.0
  (5, 0)        1.0
  (6, 0)        1.0
  (7, 0)        1.0
  (8, 0)        1.0
  (9, 0)        1.0
  (10, 1)       1.0
  (11, 0)       1.0
  (12, 1)       1.0
  (13, 0)       1.0
  (14, 1)       1.0
  (15, 0)       1.0
  (16, 0)       1.0
  (17, 0)       1.0
  (18, 0)       1.0
  (19, 0)       1.0
  (20, 0)       1.0
  (21, 0)       1.0
  (22, 0)       1.0
  (23, 0)       1.0
  (24, 0)       1.0
  :     :
  (186, 0)      1.0
  (187, 0)      1.0
  (188, 0)      1.0
  (189, 0)      1.0
  (190, 0)      1.0
  (191, 3)      1.0
  (192, 0)      1.0
  (193, 0)      1.0
```

```
(194, 0)      1.0
(195, 0)      1.0
(196, 0)      1.0
(197, 0)      1.0
(198, 0)      1.0
(199, 0)      1.0
(200, 0)      1.0
(201, 0)      1.0
(202, 0)      1.0
(203, 0)      1.0
(204, 0)      1.0
(205, 0)      1.0
(206, 0)      1.0
(207, 0)      1.0
(208, 0)      1.0
(209, 0)      1.0
(210, 0)      1.0
```

In [71]:
```
Method_Of_Fire_Control2 = ds.loc[:,['Method_Of_Fire_Control']].values
Method_Of_Fire_Control2 = Method_Of_Fire_Control2.ravel()

from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer = CountVectorizer()
Method_Of_Fire_Control2 = vectorizer.fit_transform(Method_Of_Fire_Control2)
print(Method_Of_Fire_Control2)
```

```
  (0, 3)        1
  (0, 1)        1
  (0, 4)        1
  (0, 2)        1
  (1, 3)        1
  (1, 4)        1
  (1, 6)        1
  (2, 3)        1
  (2, 1)        1
  (2, 4)        1
  (2, 2)        1
  (3, 3)        1
  (3, 1)        1
  (3, 4)        1
  (3, 2)        1
  (4, 3)        1
  (4, 1)        1
  (4, 4)        1
  (4, 2)        1
  (5, 3)        1
  (5, 1)        1
  (5, 4)        1
  (5, 2)        1
  (6, 3)        1
  (6, 1)        1
  :     :
  (204, 2)      1
  (205, 3)      1
  (205, 1)      1
  (205, 4)      1
  (205, 2)      1
  (206, 3)      1
  (206, 1)      1
```

```
(206, 4)      1
(206, 2)      1
(207, 3)      1
(207, 1)      1
(207, 4)      1
(207, 2)      1
(208, 3)      1
(208, 1)      1
(208, 4)      1
(208, 2)      1
(209, 3)      1
(209, 1)      1
(209, 4)      1
(209, 2)      1
(210, 3)      1
(210, 1)      1
(210, 4)      1
(210, 2)      1
```

Feature Reduction using Variance Threshold

In [77]:
```
from sklearn import datasets
df = datasets.load_iris(as_frame=True)
X = df.data
y = df.target
print(X)
```

```
     sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)
0                  5.1               3.5                1.4               0.2
1                  4.9               3.0                1.4               0.2
2                  4.7               3.2                1.3               0.2
3                  4.6               3.1                1.5               0.2
4                  5.0               3.6                1.4               0.2
..                 ...               ...                ...               ...
145                6.7               3.0                5.2               2.3
146                6.3               2.5                5.0               1.9
147                6.5               3.0                5.2               2.0
148                6.2               3.4                5.4               2.3
149                5.9               3.0                5.1               1.8

[150 rows x 4 columns]
```

In [78]:
```
from sklearn.feature_selection import VarianceThreshold

selector = VarianceThreshold()
selector.fit_transform(X,y)
```

Out[78]:
```
array([[5.1, 3.5, 1.4, 0.2],
       [4.9, 3. , 1.4, 0.2],
       [4.7, 3.2, 1.3, 0.2],
       [4.6, 3.1, 1.5, 0.2],
       [5. , 3.6, 1.4, 0.2],
       [5.4, 3.9, 1.7, 0.4],
       [4.6, 3.4, 1.4, 0.3],
       [5. , 3.4, 1.5, 0.2],
       [4.4, 2.9, 1.4, 0.2],
       [4.9, 3.1, 1.5, 0.1],
       [5.4, 3.7, 1.5, 0.2],
       [4.8, 3.4, 1.6, 0.2],
       [4.8, 3. , 1.4, 0.1],
       [4.3, 3. , 1.1, 0.1],
       [5.8, 4. , 1.2, 0.2],
       [5.7, 4.4, 1.5, 0.4],
       [5.4, 3.9, 1.3, 0.4],
       [5.1, 3.5, 1.4, 0.3],
       [5.7, 3.8, 1.7, 0.3],
```