

```
import pandas as pd
import numpy as np

df = pd.read_csv('/content/final_project_FEMA.csv')

#Check the first few rows of the data
df.head()

specialNeeds damagedCity damagedStateAbbreviation roofDamage foundationDamage disasterNumber householdComposition destroyed residenceType repairAmount tsaEligible grossIncome waterLevel
0 0 SARASOTA FL 0 0.0 4337.0 1.0 0.0 Condo 0.0 0.0 0.0 0.0
1 0 SULPHUR LA 0 0.0 4559.0 4.0 0.0 Apartment 0.0 0.0 55000.0 0.0
2 0 HOUSTON TX 0 0.0 4332.0 1.0 0.0 Apartment 0.0 1.0 23000.0 0.0
3 0 RICHMOND TX 0 0.0 4332.0 1.0 0.0 House/Duplex 0.0 0.0 80000.0 0.0
4 0 OVIEDO FL 0 0.0 4337.0 4.0 0.0 House/Duplex 0.0 0.0 0.0 0.0

#Get info on column types and rows
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 743079 entries, 0 to 743078
Data columns (total 13 columns):
# Column Non-Null Count Dtype
---
0 specialNeeds 743079 non-null int64
1 damagedCity 743079 non-null object
2 damagedStateAbbreviation 743079 non-null object
3 roofDamage 743079 non-null int64
4 foundationDamage 743078 non-null float64
5 disasterNumber 743078 non-null float64
6 householdComposition 743078 non-null float64
7 destroyed 743078 non-null float64
8 residenceType 743078 non-null object
9 repairAmount 743078 non-null float64
10 tsaEligible 743078 non-null float64
11 grossIncome 743078 non-null float64
12 waterLevel 743078 non-null float64
dtypes: float64(8), int64(2), object(3)
memory usage: 73.7+ MB

#Summary stats for numerical columns
df.describe()

specialNeeds roofDamage foundationDamage disasterNumber householdComposition destroyed repairAmount tsaEligible grossIncome waterLevel
count 743079.000000 743079.000000 743078.000000 743078.000000 743078.000000 743078.000000 743078.000000 743078.000000 743078.000000 743078.000000
mean 0.033926 0.036642 0.005558 4398.201454 2.258545 0.001807 366.866241 0.386600 4.936506e+04 0.850243
std 0.181040 0.187882 0.074344 108.731757 1.460657 0.042475 2214.519392 0.486971 5.947536e+05 5.518519
min 0.000000 0.000000 0.000000 4332.000000 1.000000 0.000000 0.000000 0.000000 0.000000e+00 0.000000
25% 0.000000 0.000000 0.000000 4337.000000 1.000000 0.000000 0.000000 0.000000 3.900000e+03 0.000000
50% 0.000000 0.000000 0.000000 4337.000000 2.000000 0.000000 0.000000 0.000000 2.000000e+04 0.000000
75% 0.000000 0.000000 0.000000 4393.000000 3.000000 0.000000 0.000000 1.000000 4.000000e+04 0.000000
max 1.000000 1.000000 1.000000 4611.000000 22.000000 1.000000 36000.000000 1.000000 9.900000e+07 336.000000

#Check for missing values
df.isnull().sum()

specialNeeds 0
damagedCity 0
damagedStateAbbreviation 0
roofDamage 0
foundationDamage 1
disasterNumber 1
householdComposition 1
destroyed 1
residenceType 1
repairAmount 1
tsaEligible 1
grossIncome 1
waterLevel 1

dtype: int64

# Drop rows with any missing values
df = df.dropna()

Descriptive Analysis

# Total cases where TSA was used
tsa_cases_by_state = df[df['tsaEligible'] == 1].groupby('damagedStateAbbreviation').size()
print(tsa_cases_by_state)

damagedStateAbbreviation
FL 103576
LA 18467
NC 739
PR 120662
TX 43830
dtype: int64

#Average household income where TSA was used
income_tsa_used = df[df['tsaEligible'] == 1]['grossIncome'].mean()
print(income_tsa_used)

43588.34777250987

#Average household income where TSA was not used
income_tsa_not_used = df[df['tsaEligible'] == 0]['grossIncome'].mean()
print(income_tsa_not_used)

53005.87127361761

#Average family size where TSA was used
family_size_tsa_used = df[df['tsaEligible'] == 1]['householdComposition'].mean()
print(family_size_tsa_used)

2.219598710638624

#Average family size where TSA was not used
family_size_tsa_not_used = df[df['tsaEligible'] == 0]['householdComposition'].mean()
print(family_size_tsa_not_used)
```

2.2830909777009416

```
#Assessing if water level was a factor in TSA approval
water_level_tsa = df.groupby('tsaEligible')['waterLevel'].size()
print(water_level_tsa)
```

tsaEligible
0.0 455804
1.0 287274
Name: waterLevel, dtype: int64

```
#Average cost of repairs where TSA was used
repairs_cost_tsa_used = df[df['tsaEligible'] == 1]['repairAmount'].mean()
print(repairs_cost_tsa_used)
```

587.160434567695

```
#Average cost of repairs where TSA was not used
repairs_cost_tsa_not_used = df[df['tsaEligible'] == 0]['repairAmount'].mean()
print(repairs_cost_tsa_not_used)
```

228.02411970934875

```
#Count of TSA uses by residence type
tsa_by_residence_type = df.groupby('residenceType').size()
print(tsa_by_residence_type)
```

residenceType
Apartment 143131
Assisted Living Facility 930
Boat 527
College Dorm 254
Condo 20534
Correctional Facility 77
House/Duplex 458247
Military Housing 361
Mobile Home 53359
Other 29800
Townhouse 31229
Travel Trailer 4589
Unknown 40
dtype: int64

```
#Average foundation damage by TSA eligibility
foundation_damage_by_tsa = df.groupby('tsaEligible')['foundationDamage'].mean()
print(foundation_damage_by_tsa)
```

tsaEligible
0.0 0.004072
1.0 0.007916
Name: foundationDamage, dtype: float64

```
#Average roof damage by TSA eligibility
roof_damage_by_tsa = df.groupby('tsaEligible')['roofDamage'].mean()
print(roof_damage_by_tsa)
```

tsaEligible
0 0.021370
1 0.060507
Name: roofDamage, dtype: float64

Disaster-Specific Analysis

```
#Hurricane Irma cases
hurricane_irma = df[df['disasterNumber'] == 4337]
```

```
#Total TSA cases for hurricane Irma
irma_tsa_cases = hurricane_irma['tsaEligible'].sum()
print(irma_tsa_cases)
```

```
#Average repair cost for hurricane Irma
irma_repair_cost = hurricane_irma['repairAmount'].mean()
print(irma_repair_cost)
```

102597.0
71.99544358772415

```
#Hurricane Maria cases
hurricane_maria = df[df['disasterNumber'] == 4339]
```

```
#Total TSA cases for hurricane Maria
maria_tsa_cases = hurricane_maria['tsaEligible'].sum()
print(maria_tsa_cases)
```

```
#Average repair cost for hurricane Maria
maria_repair_cost = hurricane_maria['repairAmount'].mean()
print(maria_repair_cost)
```

120662.0
493.9644923290006

```
#Hurricane Harvey cases
hurricane_harvey = df[df['disasterNumber'] == 4332]
```

```
#Total TSA cases for hurricane Harvey
harvey_tsa_cases = hurricane_harvey['tsaEligible'].sum()
print(harvey_tsa_cases)
```

```
#Average repair cost for hurricane Harvey
harvey_repair_cost = hurricane_harvey['repairAmount'].mean()
print(harvey_repair_cost)
```

43830.0
1019.2771995484376

```
#Hurricane Ida cases
hurricane_ida = df[df['disasterNumber'] == 4611]
```

```
#Total TSA cases for hurricane Ida
ida_tsa_cases = hurricane_ida['tsaEligible'].sum()
print(ida_tsa_cases)
```

```
#Average repair cost for hurricane Ida
ida_repair_cost = hurricane_ida['repairAmount'].mean()
print(ida_repair_cost)
```

18467.0
456.4087519993265

```
#Texas winter storm cases
texas_winter_storm = df[df['disasterNumber'] == 4586]
```

```
#Total TSA cases for Texas Winter Storm
texas_storm_tsa_cases = texas_winter_storm['tsaEligible'].sum()
print(texas_winter_storm)
```

```
#Average repair cost for Texas Winter Storm
texas_storm_repair_cost = texas_winter_storm['repairAmount'].mean()
print(texas_storm_repair_cost)
```

specialNeeds damagedCity damagedStateAbbreviation roofDamage \
10 0 HOUSTON TX 0
52 1 HOUSTON TX 0
53 0 GRAND PRAIRIE TX 0

```

92      0      GARLAND      TX      0
105     0      SUGAR LAND    TX      0
...     ...      ...      ...      ...
742999  0      GALENA PARK    TX      0
743048  0      PLANO          TX      0
743058  1      PEARLAND       TX      0
743069  0      HOUSTON       TX      0
743073  0      HOUSTON       TX      0

      foundationDamage  disasterNumber  householdComposition  destroyed \
10      0.0      4586.0      3.0      0.0
52      0.0      4586.0      5.0      0.0
53      0.0      4586.0      1.0      0.0
92      0.0      4586.0      4.0      0.0
105     0.0      4586.0      1.0      0.0
...     ...      ...      ...      ...
742999  0.0      4586.0      5.0      0.0
743048  0.0      4586.0      1.0      0.0
743058  0.0      4586.0      1.0      0.0
743069  0.0      4586.0      1.0      0.0
743073  0.0      4586.0      1.0      0.0

      residenceType  repairAmount  tsaEligible  grossIncome  waterLevel
10  House/Duplex      0.0      0.0      4500.0      0.0
52  House/Duplex      0.0      0.0      26260.0      0.0
53  House/Duplex      0.0      0.0      0.0      0.0
92  Apartment         0.0      0.0      44000.0      0.0
105 House/Duplex      0.0      0.0      22000.0      0.0
...     ...      ...      ...      ...
742999 House/Duplex      0.0      0.0      39000.0      0.0
743048 Apartment       0.0      0.0      43000.0      0.0
743058 House/Duplex      0.0      0.0      15000.0      0.0
743069 House/Duplex      0.0      0.0      110000.0      0.0
743073 House/Duplex      0.0      0.0      54000.0      0.0

[48690 rows x 13 columns]
182.70509468063258

```

```

#Hurricane Laura cases
hurricane_laura = df[df['disasterNumber'] == 4559]

```

```

#Total TSA cases for hurricane Laura
laura_tsa_cases = hurricane_laura['tsaEligible'].sum()
print(laura_tsa_cases)

```

```

#Average repair cost for hurricane Laura
laura_repair_cost = hurricane_laura['repairAmount'].mean()
print(laura_repair_cost)

```

```

0.0
420.9760269145706

```

```

#Hurricane Florence cases
hurricane_florence = df[df['disasterNumber'] == 4393]

```

```

#Total TSA cases for hurricane Florence
florence_tsa_cases = hurricane_florence['tsaEligible'].sum()
print(florence_tsa_cases)

```

```

#Average repair cost for hurricane Florence
florence_repair_cost = hurricane_florence['repairAmount'].mean()
print(florence_repair_cost)

```

```

739.0
516.4108257679393

```

```

#Hurricane Micheal cases
hurricane_michael = df[df['disasterNumber'] == 4399]

```

```

#Total TSA cases for hurricane Micheal
michael_tsa_cases = hurricane_michael['tsaEligible'].sum()
print(michael_tsa_cases)

```

```

#Average repair cost for hurricane Micheal
michael_repair_cost = hurricane_michael['repairAmount'].mean()
print(michael_repair_cost)

```

```

979.0
588.9958535378524

```