Homework 02

Keith Wampler

AI Tool Used: Gemini 2.5 Pro

Task 1 - NRI Data Cleaning

1. Import the NRI data. Ensure that the FIPS code variable ('STCOFIPS') is correctly identified as a string / character variable. Otherwise, the leading zeros will be removed.

In [1]:

```
import pandas as pd
import numpy as np

# path to the NRI file
nri_path = '/Users/keith/Documents/code/Intro to ML 2025/data/raw/NRI_Table_

# import the data w/ FIPS column as a string type
nri_df = pd.read_csv(nri_path, dtype={'STCOFIPS': str})

# check
print(nri_df.head())
```

```
   OID_  NRI_ID    STATE STATEABBRV  STATEFIPS   COUNTY COUNTYTYPE  \
0     1  C01001  Alabama         AL          1  Autauga     County   
1     2  C01003  Alabama         AL          1  Baldwin     County   
2     3  C01005  Alabama         AL          1  Barbour     County   
3     4  C01007  Alabama         AL          1     Bibb     County   
4     5  C01009  Alabama         AL          1   Blount     County   

   COUNTYFIPS STCOFIPS  POPULATION  ...  WNTW_EALS            WNTW_EALR  \
0           1    01001       58764  ...  15.784587             Very Low   
1           3    01003      231365  ...  56.205509  Relatively Moderate   
2           5    01005       25160  ...  18.632002       Relatively Low   
3           7    01007       22239  ...  13.308573             Very Low   
4           9    01009       58992  ...  23.645930       Relatively Low   

      WNTW_ALRB     WNTW_ALRP     WNTW_ALRA WNTW_ALR_NPCTL    WNTW_RISKV  \
0  2.687716e-07  7.410082e-09  8.725777e-06      10.461158   8494.906508   
1  1.268231e-09  2.287120e-08  1.548360e-07      13.339523  65619.701638   
2  5.788050e-07  2.347236e-08  7.606598e-07      16.125039  15501.730335   
3  9.014679e-07  1.270300e-08  1.202015e-05      16.991643   7496.186940   
4  5.268425e-07  1.482016e-08  2.002965e-07      12.039616  17175.160729   

   WNTW_RISKS      WNTW_RISKR     NRI_VER  
0   12.217626        Very Low  March 2023  
1   52.083996  Relatively Low  March 2023  
2   19.535476        Very Low  March 2023  
3   11.104041        Very Low  March 2023  
4   21.444480        Very Low  March 2023  

[5 rows x 465 columns]
```

2. Subset the NRI data to include only the 5-digit state/county FIPS code and all colums ending with '_AFREQ' and '_RISKR'. Each of these columns represents a different hazard type.

In [2]:
```python
# find columns ending with '_AFREQ' or '_RISKR'
selected_cols = ['STCOFIPS'] + [col for col in nri_df.columns if col.endswit

# create the subset
nri_subset = nri_df[selected_cols]

# check
print(nri_subset.head())
```

```
  STCOFIPS  AVLN_AFREQ      AVLN_RISKR  CFLD_AFREQ      CFLD_RISKR  \
0    01001         NaN  Not Applicable         NaN  Not Applicable   
1    01003         NaN  Not Applicable    3.684142  Relatively Low   
2    01005         NaN  Not Applicable         NaN  Not Applicable   
3    01007         NaN  Not Applicable         NaN  Not Applicable   
4    01009         NaN  Not Applicable         NaN  Not Applicable   

   CWAV_AFREQ CWAV_RISKR  DRGT_AFREQ           DRGT_RISKR  ERQK_AFREQ  ...  
\
0         0.0  No Rating   25.969774       Relatively Low    0.000431  ...   
1         0.0  No Rating   12.353442  Relatively Moderate    0.000338  ...   
2         0.0  No Rating   43.956953       Relatively Low    0.000227  ...   
3         0.0  No Rating   28.894501             Very Low    0.000790  ...   
4         0.0  No Rating   28.152598       Relatively Low    0.000817  ...   

  TRND_AFREQ           TRND_RISKR TSUN_AFREQ         TSUN_RISKR VLCN_AFREQ  
\
0   0.480184  Relatively Moderate        NaN     Not Applicable        NaN   
1   0.953140  Relatively Moderate        NaN  Insufficient Data        NaN   
2   0.739018  Relatively Moderate        NaN     Not Applicable        NaN   
3   0.586160  Relatively Moderate        NaN     Not Applicable        NaN   
4   0.710332  Relatively Moderate        NaN     Not Applicable        NaN   

       VLCN_RISKR WFIR_AFREQ           WFIR_RISKR WNTW_AFREQ      WNTW_RISKR  
0  Not Applicable   0.000035             Very Low   0.433437        Very Low  
1  Not Applicable   0.002229  Relatively Moderate   0.182759  Relatively Low  
2  Not Applicable   0.000038             Very Low   0.185759        Very Low  
3  Not Applicable   0.000040             Very Low   0.743034        Very Low  
4  Not Applicable   0.000035             Very Low   0.866873        Very Low  

[5 rows x 37 columns]
```

3. Create a table / dataframe that, for each hazard type, shows the number of missing values in the '_AFREQ' and '_RISKR' columns.

In [3]:

```python
# extract unique hazard prefixes
hazards = sorted(list(set([col.split('_')[0] for col in nri_subset.columns i

# Create a dictionary to store missing value counts
missing_counts = {}

for hazard in hazards:
    afreq_col = f"{hazard}_AFREQ"
    riskr_col = f"{hazard}_RISKR"
    missing_counts[hazard] = {
        'AFREQ_missing': nri_subset[afreq_col].isnull().sum(),
        'RISKR_missing': nri_subset[riskr_col].isnull().sum()
    }

# convert the dictionary to a df printing
missing_df = pd.DataFrame.from_dict(missing_counts, orient='index')

print(missing_df)
```

```
      AFREQ_missing  RISKR_missing
AVLN           3023              0
CFLD           2646              0
CWAV              0              0
DRGT              7              0
ERQK              0              0
HAIL              7              0
HRCN            918              0
HWAV              0              0
ISTM            229              0
LNDS             40              0
LTNG            123              0
RFLD              0              0
SWND              7              0
TRND              7              0
TSUN           3103              0
VLCN           3125              0
WFIR             88              0
WNTW              0              0
```

4. Create a new column in the original data table indicating whether or not 'AVLN_AFREQ' is missing or observed. Show the cross-tabulation of the 'AVLN_AFREQ' missingness and 'AVLN_RISKR' columns (including missing values). What do you observe?

In [4]:
```python
# create a new column that is True if AVLN_AFREQ is null, and False otherwis
nri_df['AVLN_AFREQ_missing'] = nri_df['AVLN_AFREQ'].isnull()

# make cross-tabulation table
# dropna=False ensures we see counts of missing values in the RISKR column a
cross_tab = pd.crosstab(
    nri_df['AVLN_AFREQ_missing'],
    nri_df['AVLN_RISKR'],
    dropna=False
)

print(cross_tab)
```

```
AVLN_RISKR          Not Applicable  Relatively High  Relatively Low  \
AVLN_AFREQ_missing
False                            0               15              52
True                          3023                0               0

AVLN_RISKR          Relatively Moderate  Very High  Very Low
AVLN_AFREQ_missing
False                                33          9        99
True                                  0          0         0
```

Observation: AVLN_AFREQ is missing in every single case where AVLN_RISKR is "Not Applicable". This suggests that when an avalanche risk isn't applicable to a county (e.g., Florida), its frequency isn't recorded.

5. Assuming that a risk that is “not applicable” to a county has an annualized frequency of 0, impute the relevant missing values in the ‘_AFREQ’ columns with 0.

In [5]:
```
# make list of all AFREQ columns
afreq_columns = [col for col in nri_df.columns if col.endswith('_AFREQ')]

# fill missing values in these columns with 0
nri_df[afreq_columns] = nri_df[afreq_columns].fillna(0)

# check
print(f"missing values in AVLN_AFREQ after imputation: {nri_df['AVLN_AFREQ']
```

```
missing values in AVLN_AFREQ after imputation: 0
```

Task 2 - SVI Data Cleaning

1. Import the SVI data. Ensure that the FIPS code is correctly identified as a string / character variable. Otherwise, the leading zeros will be removed.
1. Subset the SVI data to include only the following columns: ST, STATE, ST_ABBR, STCNTY, COUNTY, FIPS, LOCATION, AREA_SQMI, E_TOTPOP, EP_POV150, EP_UNEMP, EP_HBURD, EP_NOHSDP, EP_UNINSUR, EP_AGE65, EP_AGE17, EP_DISABL, EP_SNGPNT, EP_LIMENG, EP_MINRTY, EP_MUNIT, EP_MOBILE, EP_CROWD, EP_NOVEH, EP_GROUPQ, EP_NOINT, EP_AFAM, EP_HISP, EP_ASIAN, EP_AIAN, EP_NHPI, EP_TWOMORE, EP_OTHERRACE

In [6]:
```
# path to your SVI file
svi_path = '/Users/keith/Documents/code/Intro to ML 2025/data/raw/SVI_2022_U

# import the SVI data w/ FIPS is a string
svi_df = pd.read_csv(svi_path, dtype={'FIPS': str})

# columns to keep
svi_columns_to_keep = [
    'ST', 'STATE', 'ST_ABBR', 'STCNTY', 'COUNTY', 'FIPS', 'LOCATION', 'AREA_
    'E_TOTPOP', 'EP_POV150', 'EP_UNEMP', 'EP_HBURD', 'EP_NOHSDP', 'EP_UNINSU
    'EP_AGE65', 'EP_AGE17', 'EP_DISABL', 'EP_SNGPNT', 'EP_LIMENG', 'EP_MINRT
    'EP_MUNIT', 'EP_MOBILE', 'EP_CROWD', 'EP_NOVEH', 'EP_GROUPQ', 'EP_NOINT'
    'EP_AFAM', 'EP_HISP', 'EP_ASIAN', 'EP_AIAN', 'EP_NHPI', 'EP_TWOMORE', 'E
]

# make subset
svi_subset = svi_df[svi_columns_to_keep]

# check
print(svi_subset.head())
```

```
   ST    STATE ST_ABBR  STCNTY          COUNTY   FIPS  \
0   1  Alabama      AL    1001  Autauga County  01001
1   1  Alabama      AL    1003  Baldwin County  01003
2   1  Alabama      AL    1005  Barbour County  01005
3   1  Alabama      AL    1007     Bibb County  01007
4   1  Alabama      AL    1009   Blount County  01009

                  LOCATION    AREA_SQMI  E_TOTPOP  EP_POV150  ...  EP_NOVEH
\
0  Autauga County, Alabama   594.454786     58761       20.2  ...       4.0
1  Baldwin County, Alabama  1589.861817    233420       18.3  ...       2.3
2  Barbour County, Alabama   885.007619     24877       37.7  ...      11.7
3     Bibb County, Alabama   622.469286     22251       29.0  ...       7.5
4   Blount County, Alabama   644.890376     59077       22.9  ...       4.8

   EP_GROUPQ  EP_NOINT  EP_AFAM  EP_HISP  EP_ASIAN  EP_AIAN  EP_NHPI  \
0        0.9      10.9     19.6      3.2       1.1      0.1      0.0
1        1.5      10.9      8.3      4.8       0.9      0.2      0.0
2       12.0      31.8     46.9      4.8       0.5      0.3      0.0
3        6.4      20.2     20.7      2.9       0.3      0.1      0.0
4        1.0      16.9      1.2      9.7       0.2      0.1      0.2

   EP_TWOMORE  EP_OTHERRACE
0         3.3           0.2
1         3.1           0.4
2         1.8           1.2
3         1.7           0.1
4         2.8           0.1

[5 rows x 33 columns]
```

2. Create a table / dataframe that shows the number of missing values in each column. (Hint: if you wrote a function for Task 1, you can reuse it here.)

In [7]:
```python
# sum of nulls for each column
missing_svi = svi_subset.isnull().sum()

# check
print(missing_svi)
```

```
ST              0
STATE           0
ST_ABBR         0
STCNTY          0
COUNTY          0
FIPS            0
LOCATION        0
AREA_SQMI       0
E_TOTPOP        0
EP_POV150       0
EP_UNEMP        0
EP_HBURD        0
EP_NOHSDP       0
EP_UNINSUR      0
EP_AGE65        0
EP_AGE17        0
EP_DISABL       0
EP_SNGPNT       0
EP_LIMENG       0
EP_MINRTY       0
EP_MUNIT        0
EP_MOBILE       0
EP_CROWD        0
EP_NOVEH        0
EP_GROUPQ       0
EP_NOINT        0
EP_AFAM         0
EP_HISP         0
EP_ASIAN        0
EP_AIAN         0
EP_NHPI         0
EP_TWOMORE      0
EP_OTHERRACE    0
dtype: int64
```

Task 3 - Data Merging

1. Identify any FIPS codes that are present in the NRI data but not in the SVI data and vice versa. Describe any discrepancies and possible causes? What to these discrepancies, if any, mean for interpreting results based on the merged dataset moving forward?

In [8]:
```
# get the set of FIPS codes from each df
nri_fips = set(nri_df['STCOFIPS'])
svi_fips = set(svi_df['FIPS'])

# find FIPS codes in NRI but not in SVI
nri_only = nri_fips - svi_fips
print(f"FIPS codes in NRI only: {nri_only}")
print(f"Count: {len(nri_only)}\n")


# find FIPS codes in SVI but not in NRI
svi_only = svi_fips - nri_fips
```

```
print(f"FIPS codes in SVI only: {svi_only}")
print(f"Count: {len(svi_only)}")
```

```
FIPS codes in NRI only: {'72111', '72143', '78030', '72131', '72043', '7214
7', '72087', '72031', '72141', '72065', '72035', '72049', '09013', '72097',
'69120', '72139', '69110', '72013', '72115', '72119', '72095', '72051', '600
10', '09003', '72007', '72091', '72023', '60050', '09005', '72027', '72113',
'72081', '09011', '72015', '72089', '72083', '78020', '09015', '72011', '721
35', '72075', '72117', '72009', '72061', '72071', '72099', '72133', '72077',
'72107', '72037', '72101', '78010', '09007', '72017', '72103', '72021', '721
45', '72039', '72073', '72129', '72127', '72093', '72121', '72079', '72105',
'72109', '72033', '72019', '72137', '09009', '72057', '72054', '72067', '720
25', '72151', '66010', '72125', '72153', '72047', '72123', '72059', '60020',
'69100', '72055', '72085', '09001', '72029', '72063', '72069', '72053', '721
49', '72001', '72005', '72003', '72041', '72045'}
Count: 96

FIPS codes in SVI only: {'09120', '09160', '09140', '09180', '09110', '0917
0', '09150', '09190', '09130'}
Count: 9
```

The two datasets don't perfectly align because the SVI data includes Puerto Rico while the NRI does not, and the NRI also uses a few outdated county codes. When these files are merged, you get incomplete rows for mismatched counties, and data for specific locations like military bases is simply included within their surrounding county's record. This means your results could be misleading unless you first harmonize the county codes and account for these geographic differences in your analysis.

2. Merge the NRI and SVI data on the FIPS code. Use an outer join to keep all counties in the final dataset.

In [9]:
```
# rename columns before merging
nri_cleaned = nri_df.rename(columns={'STCOFIPS': 'FIPS'})
svi_cleaned = svi_subset # Already subsetted in Task 2

# outer merge
merged_df = pd.merge(
    nri_cleaned,
    svi_cleaned,
    on='FIPS',
    how='outer'
)

# show size of merged df
print(merged_df.shape)
```

```
(3240, 498)
```

3. Create a table / dataframe that shows the number of missing values in each column of the merged dataset.

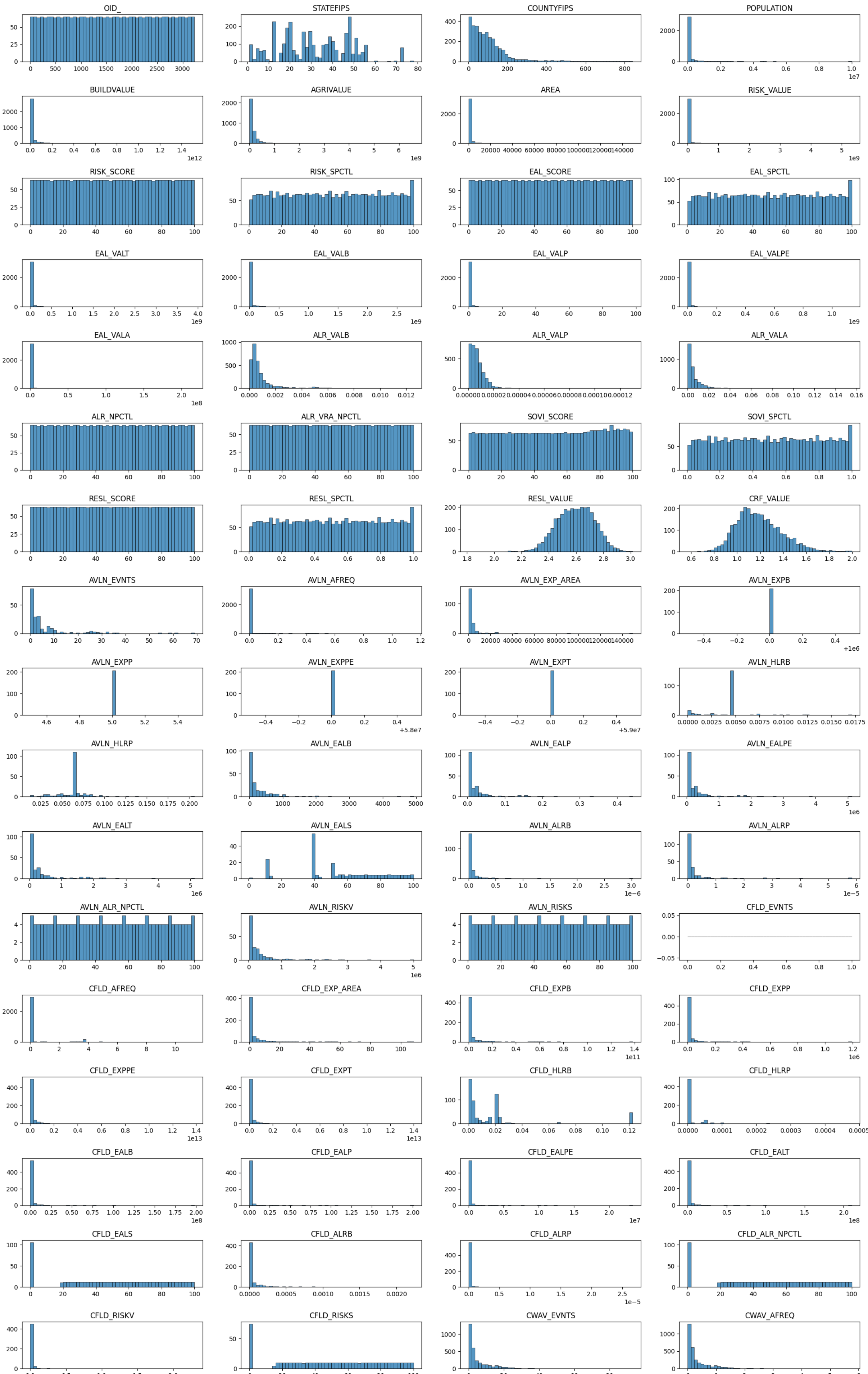
In [10]:
```
# get missing value counts for the merged df
print(merged_df.isnull().sum())
```

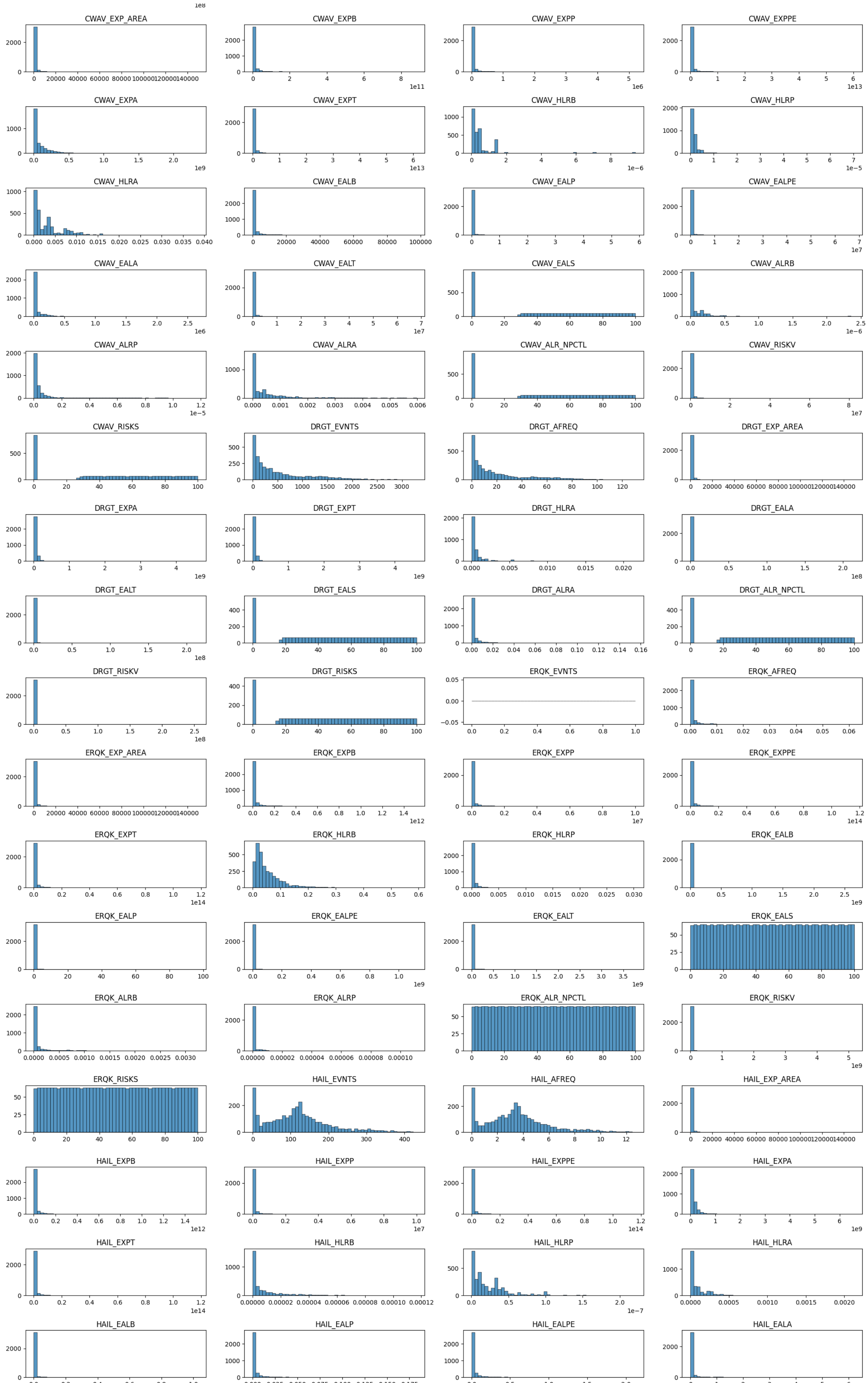
```
OID_            9
NRI_ID          9
STATE_x         9
STATEABBRV      9
STATEFIPS       9
               ..
EP_ASIAN       96
EP_AIAN        96
EP_NHPI        96
EP_TWOMORE     96
EP_OTHERRACE   96
Length: 498, dtype: int64
```

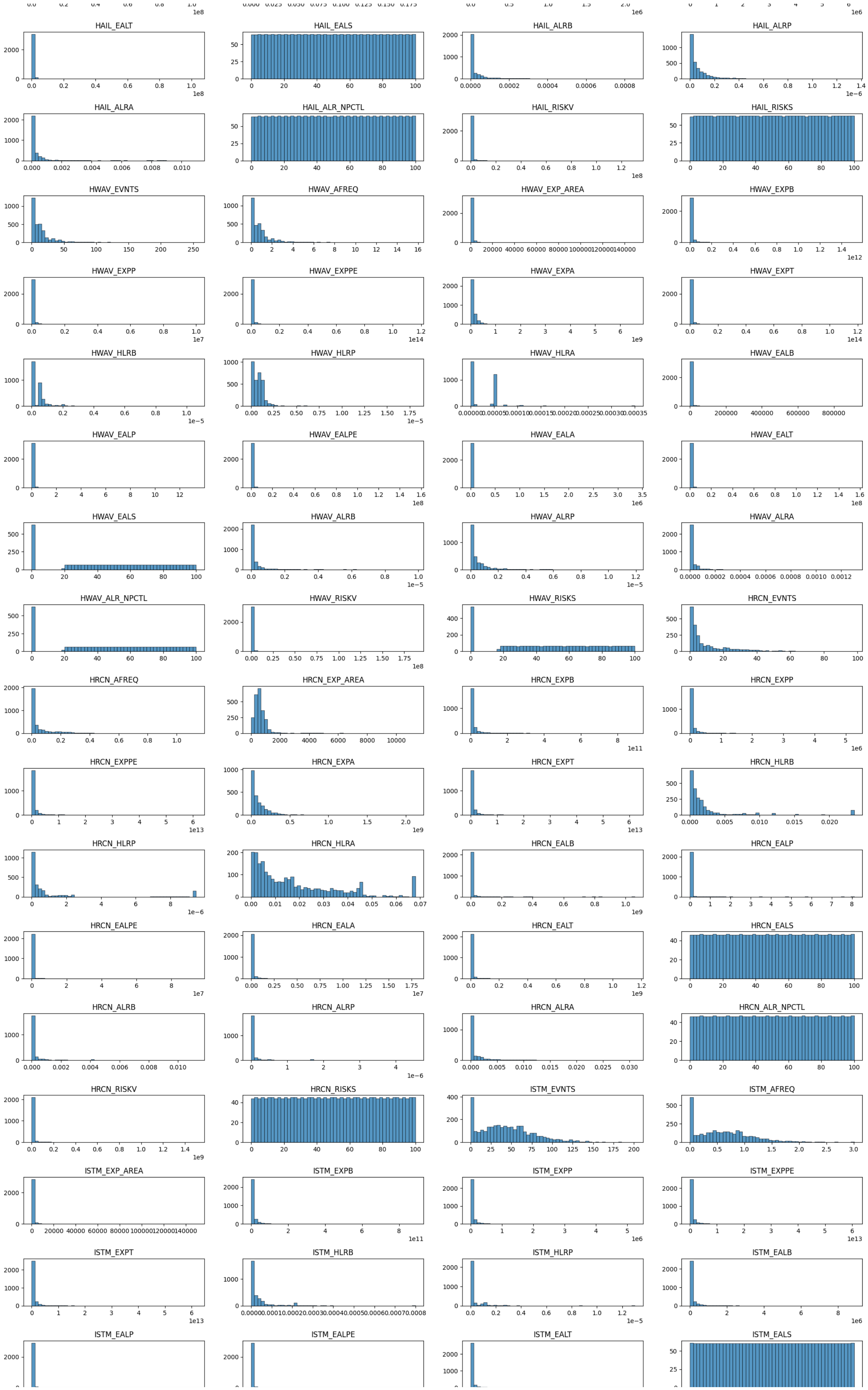
Task 4 - Data Analysis

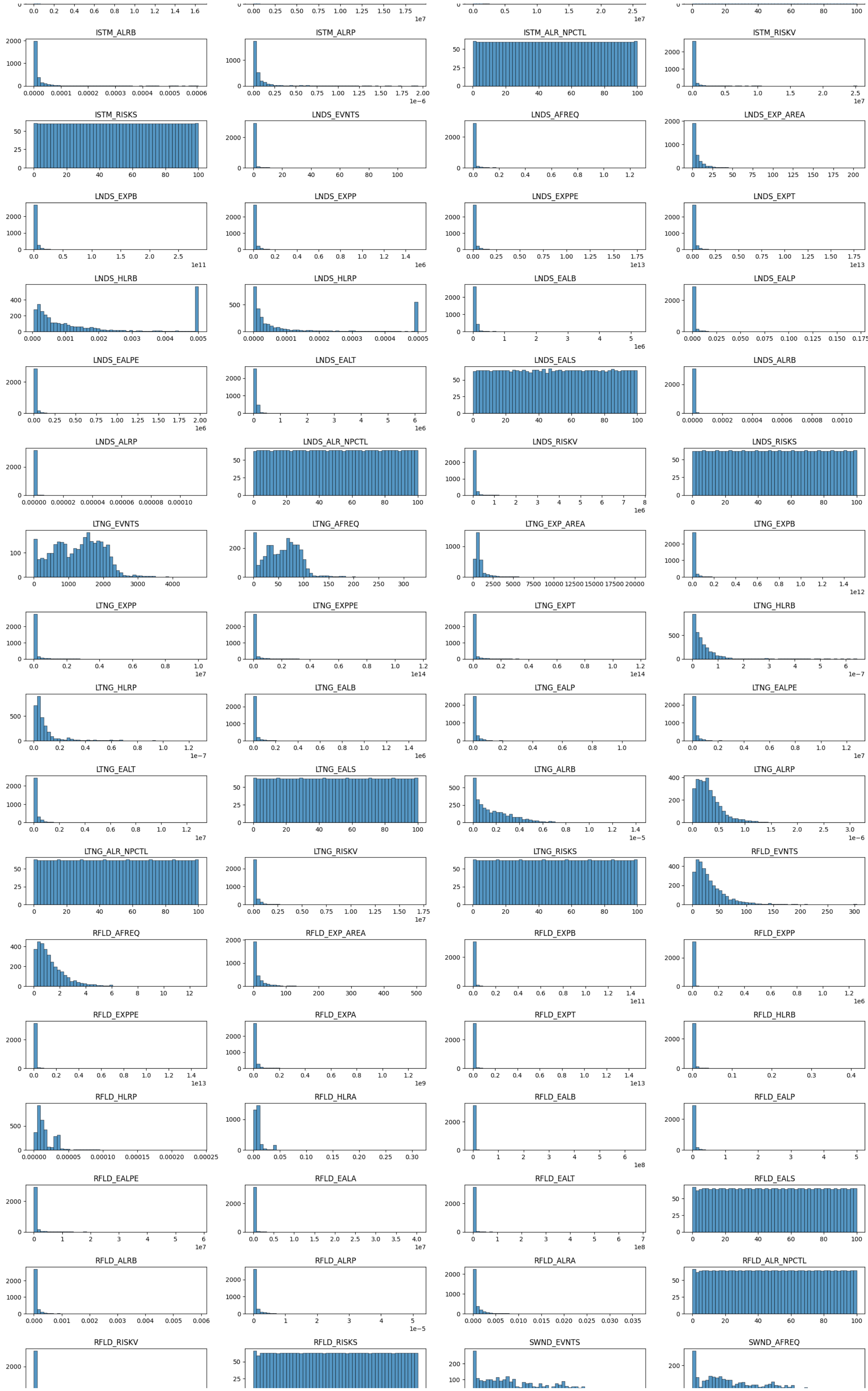
1. For each numerical variable in the merged dataset, plot a histogram showing the distribution of values. (Hint: write a function to make the histogram for a single variable, then use a loop or apply function to make the histograms for all numerical variables.)

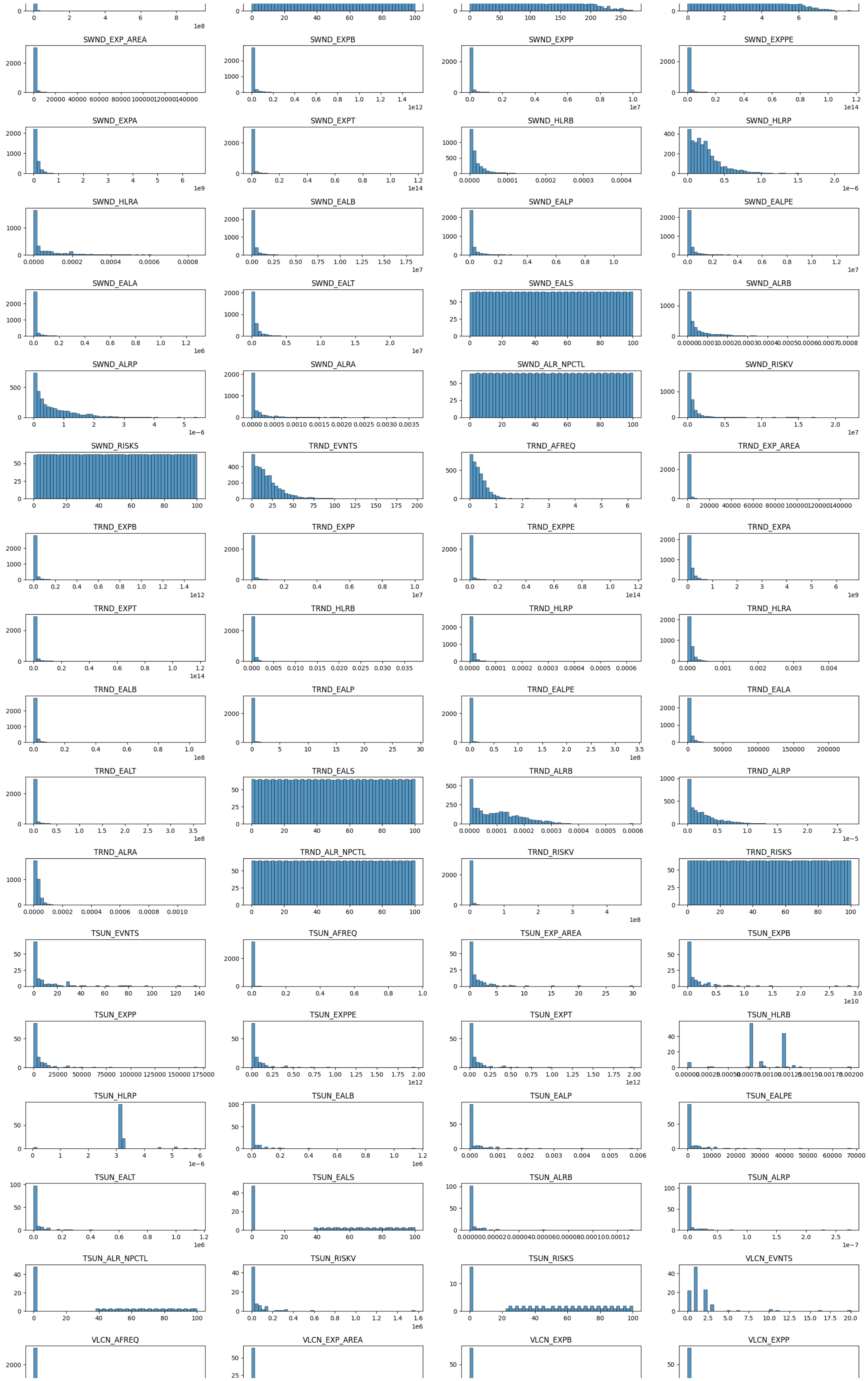
In [11]:
```python
import matplotlib.pyplot as plt
import seaborn as sns

# select only numerical columns for plotting
numerical_cols = merged_df.select_dtypes(include=np.number).columns

# figure to hold the plots
plt.figure(figsize=(20, 200))

for i, col in enumerate(numerical_cols):
    plt.subplot(len(numerical_cols) // 4 + 1, 4, i + 1)
    sns.histplot(merged_df[col], kde=False, bins=50)
    plt.title(col)
    plt.xlabel('')
    plt.ylabel('')

plt.tight_layout()
plt.show()
```

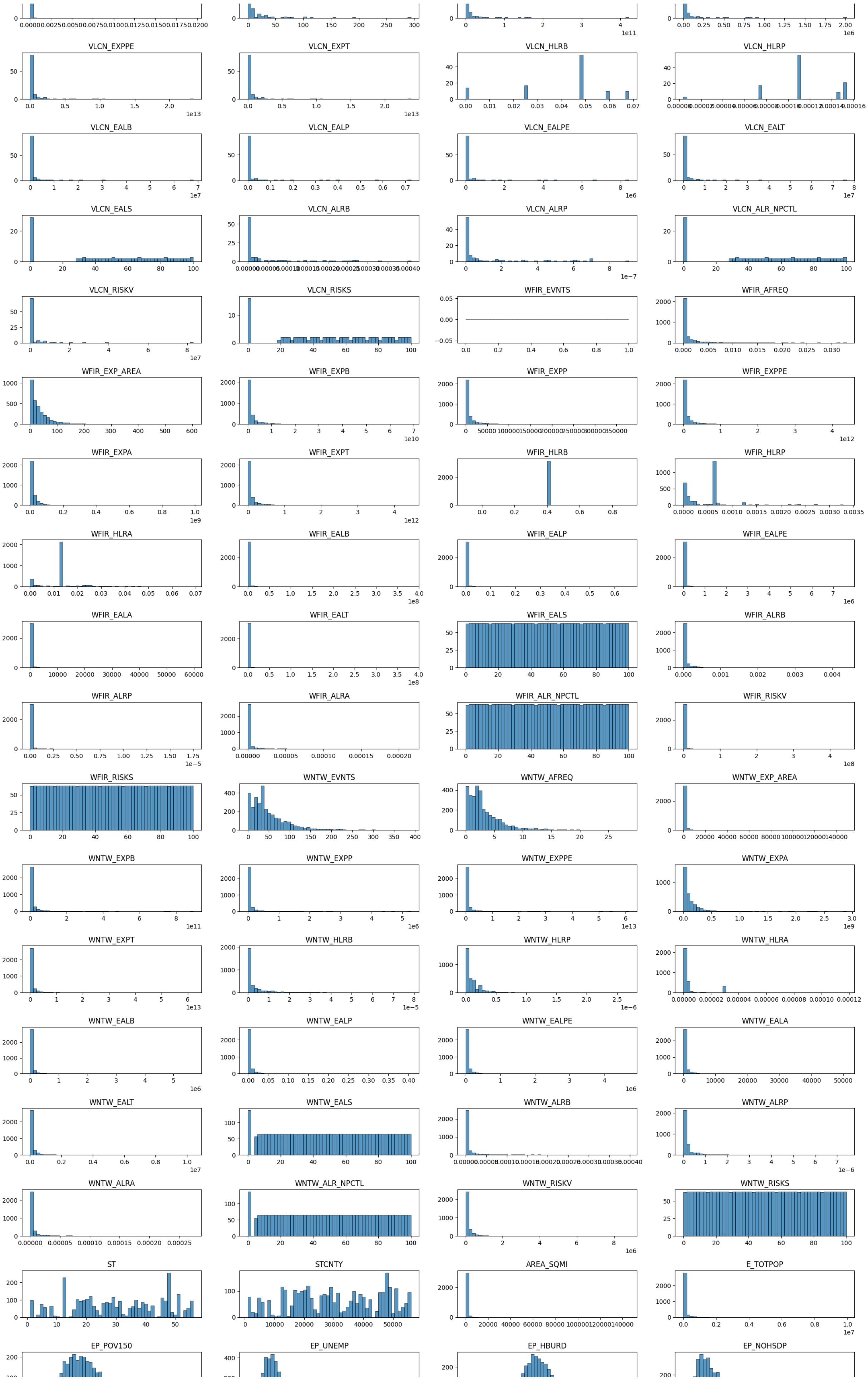

OID_
STATEFIPS
COUNTYFIPS
POPULATION
BUILDVALUE
AGRIVALUE
AREA
RISK_VALUE
RISK_SCORE
RISK_SPCTL
EAL_SCORE
EAL_SPCTL
EAL_VALT
EAL_VALB
EAL_VALP
EAL_VALPE
EAL_VALA
ALR_VALB
ALR_VALP
ALR_VALA
ALR_NPCTL
ALR_VRA_NPCTL
SOVI_SCORE
SOVI_SPCTL
RESL_SCORE
RESL_SPCTL
RESL_VALUE
CRF_VALUE
AVLN_EVNTS
AVLN_AFREQ
AVLN_EXP_AREA
AVLN_EXPB
AVLN_EXPP
AVLN_EXPPE
AVLN_EXPT
AVLN_HLRB
AVLN_HLRP
AVLN_EALB
AVLN_EALP
AVLN_EALPE
AVLN_EALT
AVLN_EALS
AVLN_ALRB
AVLN_ALRP
AVLN_ALR_NPCTL
AVLN_RISKV
AVLN_RISKS
CFLD_EVNTS
CFLD_AFREQ
CFLD_EXP_AREA
CFLD_EXPB
CFLD_EXPP
CFLD_EXPPE
CFLD_EXPT
CFLD_HLRB
CFLD_HLRP
CFLD_EALB
CFLD_EALP
CFLD_EALPE
CFLD_EALT
CFLD_EALS
CFLD_ALRB
CFLD_ALRP
CFLD_ALR_NPCTL
CFLD_RISKV
CFLD_RISKS
CWAV_EVNTS
CWAV_AFREQ

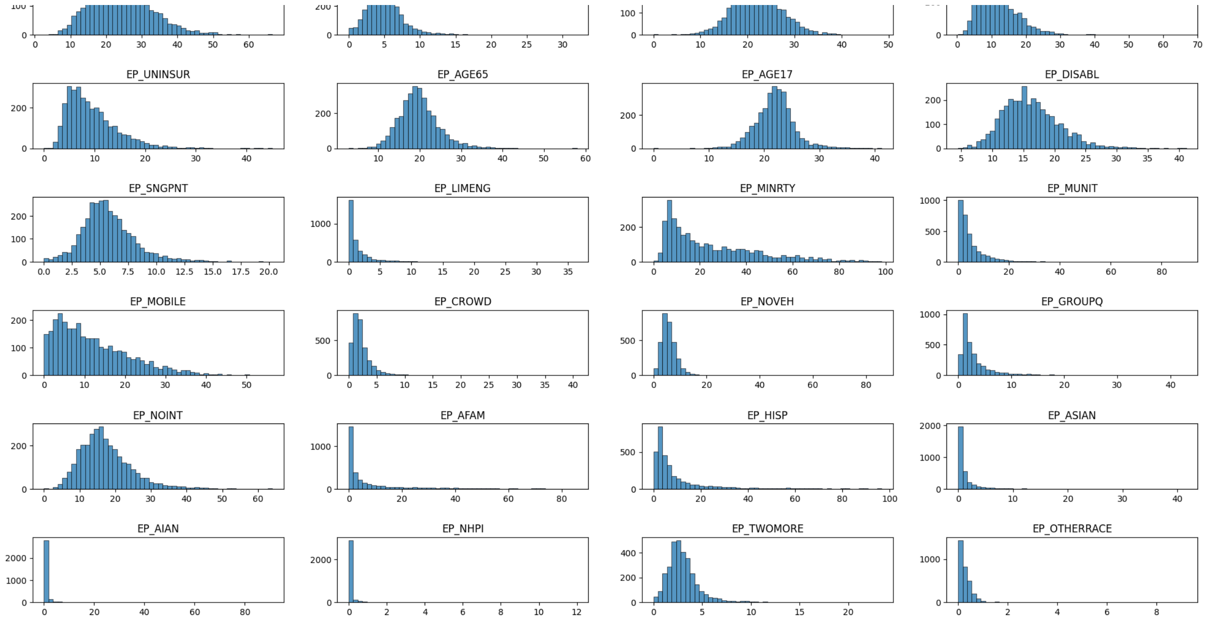

CWAV_EXP_AREA
CWAV_EXPB
CWAV_EXPP
CWAV_EXPPE
CWAV_EXPA
CWAV_EXPT
CWAV_HLRB
CWAV_HLRP
CWAV_HLRA
CWAV_EALB
CWAV_EALP
CWAV_EALPE
CWAV_EALA
CWAV_EALT
CWAV_EALS
CWAV_ALRB
CWAV_ALRP
CWAV_ALRA
CWAV_ALR_NPCTL
CWAV_RISKV
CWAV_RISKS
DRGT_EVNTS
DRGT_AFREQ
DRGT_EXP_AREA
DRGT_EXPA
DRGT_EXPT
DRGT_HLRA
DRGT_EALA
DRGT_EALT
DRGT_EALS
DRGT_ALRA
DRGT_ALR_NPCTL
DRGT_RISKV
DRGT_RISKS
ERQK_EVNTS
ERQK_AFREQ
ERQK_EXP_AREA
ERQK_EXPB
ERQK_EXPP
ERQK_EXPPE
ERQK_EXPT
ERQK_HLRB
ERQK_HLRP
ERQK_EALB
ERQK_EALP
ERQK_EALPE
ERQK_EALT
ERQK_EALS
ERQK_ALRB
ERQK_ALRP
ERQK_ALR_NPCTL
ERQK_RISKV
ERQK_RISKS
HAIL_EVNTS
HAIL_AFREQ
HAIL_EXP_AREA
HAIL_EXPB
HAIL_EXPP
HAIL_EXPPE
HAIL_EXPA
HAIL_EXPT
HAIL_HLRB
HAIL_HLRP
HAIL_HLRA
HAIL_EALB
HAIL_EALP
HAIL_EALPE
HAIL_EALA


HAIL_EALT
HAIL_EALS
HAIL_ALRB
HAIL_ALRP
HAIL_ALRA
HAIL_ALR_NPCTL
HAIL_RISKV
HAIL_RISKS
HWAV_EVNTS
HWAV_AFREQ
HWAV_EXP_AREA
HWAV_EXPB
HWAV_EXPP
HWAV_EXPPE
HWAV_EXPA
HWAV_EXPT
HWAV_HLRB
HWAV_HLRP
HWAV_HLRA
HWAV_EALB
HWAV_EALP
HWAV_EALPE
HWAV_EALA
HWAV_EALT
HWAV_EALS
HWAV_ALRB
HWAV_ALRP
HWAV_ALRA
HWAV_ALR_NPCTL
HWAV_RISKV
HWAV_RISKS
HRCN_EVNTS
HRCN_AFREQ
HRCN_EXP_AREA
HRCN_EXPB
HRCN_EXPP
HRCN_EXPPE
HRCN_EXPA
HRCN_EXPT
HRCN_HLRB
HRCN_HLRP
HRCN_HLRA
HRCN_EALB
HRCN_EALP
HRCN_EALPE
HRCN_EALA
HRCN_EALT
HRCN_EALS
HRCN_ALRB
HRCN_ALRP
HRCN_ALRA
HRCN_ALR_NPCTL
HRCN_RISKV
HRCN_RISKS
ISTM_EVNTS
ISTM_AFREQ
ISTM_EXP_AREA
ISTM_EXPB
ISTM_EXPP
ISTM_EXPPE
ISTM_EXPT
ISTM_HLRB
ISTM_HLRP
ISTM_EALB
ISTM_EALP
ISTM_EALPE
ISTM_EALT
ISTM_EALS


ISTM_ALRB
ISTM_ALRP
ISTM_ALR_NPCTL
ISTM_RISKV
ISTM_RISKS
LNDS_EVNTS
LNDS_AFREQ
LNDS_EXP_AREA
LNDS_EXPB
LNDS_EXPP
LNDS_EXPPE
LNDS_EXPT
LNDS_HLRB
LNDS_HLRP
LNDS_EALB
LNDS_EALP
LNDS_EALPE
LNDS_EALT
LNDS_EALS
LNDS_ALRB
LNDS_ALRP
LNDS_ALR_NPCTL
LNDS_RISKV
LNDS_RISKS
LTNG_EVNTS
LTNG_AFREQ
LTNG_EXP_AREA
LTNG_EXPB
LTNG_EXPP
LTNG_EXPPE
LTNG_EXPT
LTNG_HLRB
LTNG_HLRP
LTNG_EALB
LTNG_EALP
LTNG_EALPE
LTNG_EALT
LTNG_EALS
LTNG_ALRB
LTNG_ALRP
LTNG_ALR_NPCTL
LTNG_RISKV
LTNG_RISKS
RFLD_EVNTS
RFLD_AFREQ
RFLD_EXP_AREA
RFLD_EXPB
RFLD_EXPP
RFLD_EXPPE
RFLD_EXPA
RFLD_EXPT
RFLD_HLRB
RFLD_HLRP
RFLD_HLRA
RFLD_EALB
RFLD_EALP
RFLD_EALPE
RFLD_EALA
RFLD_EALT
RFLD_EALS
RFLD_ALRB
RFLD_ALRP
RFLD_ALRA
RFLD_ALR_NPCTL
RFLD_RISKV
RFLD_RISKS
SWND_EVNTS
SWND_AFREQ


SWND_EXP_AREA
SWND_EXPB
SWND_EXPP
SWND_EXPPE
SWND_EXPA
SWND_EXPT
SWND_HLRB
SWND_HLRP
SWND_HLRA
SWND_EALB
SWND_EALP
SWND_EALPE
SWND_EALA
SWND_EALT
SWND_EALS
SWND_ALRB
SWND_ALRP
SWND_ALRA
SWND_ALR_NPCTL
SWND_RISKV
SWND_RISKS
TRND_EVNTS
TRND_AFREQ
TRND_EXP_AREA
TRND_EXPB
TRND_EXPP
TRND_EXPPE
TRND_EXPA
TRND_EXPT
TRND_HLRB
TRND_HLRP
TRND_HLRA
TRND_EALB
TRND_EALP
TRND_EALPE
TRND_EALA
TRND_EALT
TRND_EALS
TRND_ALRB
TRND_ALRP
TRND_ALRA
TRND_ALR_NPCTL
TRND_RISKV
TRND_RISKS
TSUN_EVNTS
TSUN_AFREQ
TSUN_EXP_AREA
TSUN_EXPB
TSUN_EXPP
TSUN_EXPPE
TSUN_EXPT
TSUN_HLRB
TSUN_HLRP
TSUN_EALB
TSUN_EALP
TSUN_EALPE
TSUN_EALT
TSUN_EALS
TSUN_ALRB
TSUN_ALRP
TSUN_ALR_NPCTL
TSUN_RISKV
TSUN_RISKS
VLCN_EVNTS
VLCN_AFREQ
VLCN_EXP_AREA
VLCN_EXPB
VLCN_EXPP


VLCN_EXPPE
VLCN_EXPT
VLCN_HLRB
VLCN_HLRP
VLCN_EALB
VLCN_EALP
VLCN_EALPE
VLCN_EALT
VLCN_EALS
VLCN_ALRB
VLCN_ALRP
VLCN_ALR_NPCTL
VLCN_RISKV
VLCN_RISKS
WFIR_EVNTS
WFIR_AFREQ
WFIR_EXP_AREA
WFIR_EXPB
WFIR_EXPP
WFIR_EXPPE
WFIR_EXPA
WFIR_EXPT
WFIR_HLRB
WFIR_HLRP
WFIR_HLRA
WFIR_EALB
WFIR_EALP
WFIR_EALPE
WFIR_EALA
WFIR_EALT
WFIR_EALS
WFIR_ALRB
WFIR_ALRP
WFIR_ALRA
WFIR_ALR_NPCTL
WFIR_RISKV
WFIR_RISKS
WNTW_EVNTS
WNTW_AFREQ
WNTW_EXP_AREA
WNTW_EXPB
WNTW_EXPP
WNTW_EXPPE
WNTW_EXPA
WNTW_EXPT
WNTW_HLRB
WNTW_HLRP
WNTW_HLRA
WNTW_EALB
WNTW_EALP
WNTW_EALPE
WNTW_EALA
WNTW_EALT
WNTW_EALS
WNTW_ALRB
WNTW_ALRP
WNTW_ALRA
WNTW_ALR_NPCTL
WNTW_RISKV
WNTW_RISKS
ST
STCNTY
AREA_SQMI
E_TOTPOP
EP_POV150
EP_UNEMP
EP_HBURD
EP_NOHSDP


EP_UNINSUR
EP_AGE65
EP_AGE17
EP_DISABL
EP_SNGPNT
EP_LIMENG
EP_MINRTY
EP_MUNIT
EP_MOBILE
EP_CROWD
EP_NOVEH
EP_GROUPQ
EP_NOINT
EP_AFAM
EP_HISP
EP_ASIAN
EP_AIAN
EP_NHPI
EP_TWOMORE
EP_OTHERRACE