```
"""
Script to process and aggregate FEMA flood damage data sets for input to
GIS software. This script is part of the guidance for flood mapping
to inform community-led buyout and relocations projects.
Author: Michelle Hummel, Ph.D., University of Texas at Arlington
Email: michelle.hummel@uta.edu
"""

import pandas as pd
import numpy as np
import os

##########
#SET FILE LOCATIONS
#UNCOMMENT THIS SECTION AFTER ADDING FILE PATHS/NAMES
##########

#data_dir = <path to folder containing data>
#os.chdir(data_dir)
#
#zipListFile = <csv file listing zip codes (5 digit codes) in study region>
#ctListFile = <csv file listing Census tracts (11 digit codes) in study region>
#cbListFile = <csv file listing Census blocks (15 digit codes) in study region>
#nfipFile = <csv file with NFIP claims from Ref 1>
#srlpFile = <csv file with SRLP claims from NRDC>
#iaFile = <csv file with IA claims from Ref 2>
#
#nfipOut = <txt file for NFIP outputs>
#srlpOut = <txt file for SRLP outputs>
#iaOut = <txt file for IA outputs>

##########
#IMPORT DATA
##########

zipList = pd.read_csv(zipListFile,header=None,names=['zip'],dtype='int64')
ctList = pd.read_csv(ctListFile,header=None,names=['censusTract'],dtype='int64')
cbList = pd.read_csv(cbListFile,header=None,names=['censusBlock'],dtype='int64')
nfip=pd.read_csv(nfipFile)
srlp = pd.read_csv(srlpFile)
ia = pd.read_csv(iaFile)

##########
#MODIFY NFIP DATA
##########

nfip['claimcount'] = np.ones(nfip.shape[0])
nfip_head = nfip.head()

#group NFIP claims by census tract
byTract = pd.pivot_table(nfip,index='censustract',\
                         values=['amountpaidonbuildingclaim',\
                                 'amountpaidoncontentsclaim',\
                                 'totalbuildinginsurancecoverage',\
                                 'totalcontentsinsurancecoverage',\
                                 'yearofloss','claimcount'],\
```

```
                        aggfunc={'amountpaidonbuildingclaim':np.sum,\
                                 'amountpaidoncontentsclaim':np.sum,\
                                 'totalbuildinginsurancecoverage':np.mean,\
                                 'totalcontentsinsurancecoverage':np.mean,\
                                 'yearofloss':np.mean,'claimcount':np.sum})

#select values in desired census tracts
byTract['tractID'] = byTract.index
byTractFiltered = byTract[byTract.tractID.isin(ctList.censusTract)]

#save to table for input into ArcGIS
byTractFiltered.to_csv(nfipOut)

##########
#MODIFY SLRP DATA
##########

srlp_head = srlp.head()

#group SLRP data by zip code
byZip = pd.pivot_table(srlp,index='5-digit zip',\
                        values=['Tot Building Payment','Tot Contents Payment',\
                                'Losses','Total Paid','Average Pay',\
                                'Building Value'],\
                        aggfunc={'Tot Building Payment':np.mean,\
                                 'Tot Contents Payment':np.mean,\
                                 'Losses':np.mean,'Total Paid':np.mean,\
                                 'Average Pay':np.mean,'Building Value':np.mean})

#select values in desired zip codes
byZip['zipID'] = byZip.index
byZipFiltered = byZip[byZip.zipID.isin(zipList.zip)]

#save to table for input into ArcGIS
byZipFiltered.to_csv(srlpOut)

##########
#INDIVIDUAL ASSISTANCE DATA
##########

ia['claimcount'] = np.ones(ia.shape[0])
ia['ownRentKey'] = np.zeros(ia.shape[0])
ia.loc[ia.ownRent =='Renter','ownRentKey'] = 1
ia_head = ia.head()

#group IA claims by census block
byBlock = pd.pivot_table(ia_head,index='censusBlockId',\
                        values=['floodInsurance','waterLevel',\
                                'claimcount','grossIncome','ownRentKey'],\
                        aggfunc={'floodInsurance':np.sum,'waterLevel':np.mean,\
                                 'claimcount':np.sum,'grossIncome':np.mean,\
                                 'ownRentKey':np.sum})

#select values in desired census blocks
byBlock['blockID'] = byBlock.index
byBlockFiltered = byBlock[byBlock.blockID.isin(cbList.censusBlock)]
```

```
#save to table for input into ArcGIS
byBlockFiltered.to_csv(iaOut)
```