Preparing demolition data by the City of Austin

- We start with Construction permits (https://data.austintexas.gov/Building-and-Development/Issued-Construction-Permits/3syk-w9eu/data) published on the city's Socrata data portal.
- We've flitered it to permits for full building demolitions (https://data.austintexas.gov/Building-and-Development/demolitions-full-post2017/4d8v-cjdw) and for partial demos (https://data.austintexas.gov/Building-and-Development/demolitions-partial-post2007/8qw5-9tag) based on conversations with permitting folks at the City of Austin. Notes in the README of the github repo.
- We combine those two files, apply some global filtering to get active/complete residential permits and to clip permits newer than June 30, 2018.
- We then split the permit types again to apply filters specific to full or partial demos.
- We export the data gain for analysis in other notebooks.

We do this work in a separate workbook to save time and confusion, so downloading and processing can be done outside of analysis.

Set up and configurations

In [1]:

```
import pandas as pd
```

Download file from Socrata

This downloads files directly from Socrata. They are the filtered views saved and noted above.

In [2]:

```
%%bash
curl -L -o ../data-raw/full-downloaded.csv \
https://data.austintexas.gov/resource/4d8v-cjdw.csv?\$limit=10000
curl -L -o ../data-raw/partials-downloaded.csv \
https://data.austintexas.gov/resource/8qw5-9tag.csv?\$limit=10000
```

```
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Curre
nt
                                 Dload  Upload   Total   Spent    Left  Speed
100 5829k    0 5829k    0     0   735k      0 --:--:--  0:00:07 --:--:--  827
k
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Curre
nt
                                 Dload  Upload   Total   Spent    Left  Speed
100 5936k    0 5936k    0     0  1202k      0 --:--:--  0:00:04 --:--:-- 1310
k
```

Import and processing

Some of the import configurations are used with both files.

In [3]:

```
# Column data type fixes
column_types = {
    "ApplicantPhone": pd.np.str,
    "ContractorPhone": pd.np.str,
    'CalendarYearIssued': pd.np.str,
}
```

In [4]:

```
# import full demolitions data
raw_full = pd.read_csv(
    '../data-raw/full-downloaded.csv',
    index_col=None,
    dtype=column_types,
    parse_dates=['IssuedDate']
)

raw_full.shape
```

Out[4]:

```
(7930, 66)
```

In [5]:

```
# import partials data
raw_partial = pd.read_csv(
    '../data-raw/partials-downloaded.csv',
    index_col=None,
    dtype=column_types,
    parse_dates=['IssuedDate']
)

raw_partial.shape
```

Out[5]:

```
(6347, 66)
```

Combine full and partial demos

In [6]:

```
data_raw = pd.concat([raw_full,raw_partial])

data_raw.shape
```

Out[6]:

```
(14277, 66)
```

Global filters

For any filters that might apply to both full and partial permits. Basically weeding out commercial permits, and those of unwanted status.

Create Partial vs Full flag

Since we are working with both full and partial demolitions, we need a flag to separate them so we can apply filters later that are specific to each.

In [7]:

```
# function to determine type
def set_demo_type(row):
    """ Function to evaluate demolition type. """

    if row['WorkClass'] == 'Demolition':
        return 'Full'
    else:
        return 'Partial'

# apply get_winner function to new column through .assign
data_typed = data_raw.assign(
    DemoType=data_raw.apply(
        set_demo_type,
        axis=1
        )
)

# peek at it
data_typed.DemoType.value_counts()
```

Out[7]:

```
Full       7930
Partial    6347
Name: DemoType, dtype: int64
```

Filter for permit status

We want only Active and Final, per interview with the city.

In [8]:

```
# Consider permit status.
data_typed.StatusCurrent.value_counts()
```

Out[8]:

```
Final                              9730
Expired                            2067
Active                             1634
VOID                                612
Withdrawn                           214
Inactive Pending Revision            10
Aborted                               4
Closed                                2
On Hold                               2
Cancelled - Contractor Required       2
Name: StatusCurrent, dtype: int64
```

In [9]:
```
# We want only Active and Final, per city
permit_status_filtered = data_typed[
    (data_typed['StatusCurrent'] == 'Active')
    | (data_typed['StatusCurrent'] == 'Final')
]
permit_status_filtered.shape
```

Out[9]:
```
(11364, 67)
```

Filter for residential vs commercial

In [10]:
```
# show residential vs commercial
permit_status_filtered.PermitClassMapped.value_counts()
```

Out[10]:
```
Residential    10547
Commercial       817
Name: PermitClassMapped, dtype: int64
```

In [11]:
```
# filter to just residential premits
filtered_residential = permit_status_filtered[
    permit_status_filtered['PermitClassMapped'] == 'Residential'
]
filtered_residential.shape
```

Out[11]:
```
(10547, 67)
```

Cut off June 30, 2018

In [12]:
```
# filter by date to set at half the year
cutoff = filtered_residential[filtered_residential['IssuedDate'] < '2018-07-01']
cutoff.shape
```

Out[12]:
```
(10489, 67)
```

Set demolitions_all dataframe and export

In [13]:
```
# Set final dataframe for analysis
demolitions_cut = cutoff
```

Filters specific to full demolitions

We filter the full demolitions by one- and two-family homes to remove garages, etc.

In [14]:

```
# Look at Permit class to spot those the are full demos
demolitions_cut['PermitClass'].value_counts().sort_index()
```

Out[14]:

```
R- 101 Single Family Houses                 15
R- 102 Secondary Apartment                   9
R- 103 Two Family Bldgs                      2
R- 329 Res Structures Other Than Bldg       20
R- 330  Accessory Use to Primary            18
R- 434 Addition & Alterations             5059
R- 435 Renovations/Remodel                 588
R- 436 Addn to increase housing units        1
R- 437 Residential Boat Dock                 1
R- 438 Residential Garage/Carport Addn      11
R- 645 Demolition One Family Homes        3039
R- 646 Demolition Two Family Bldgs         121
R- 649 Demolition All Other Bldgs Res     1605
Name: PermitClass, dtype: int64
```

In [15]:

```
# Filter to only full demo homes people live in
filtered_homes = demolitions_cut[
    (demolitions_cut['PermitClass'] == "R- 645 Demolition One Family Homes")
    | (demolitions_cut['PermitClass'] == "R- 646 Demolition Two Family Bldgs")
]

# peek at the results
filtered_homes.PermitClass.value_counts()
```

Out[15]:

```
R- 645 Demolition One Family Homes    3039
R- 646 Demolition Two Family Bldgs     121
Name: PermitClass, dtype: int64
```

Set Full demolitions dataframe and export

In [16]:

```
demolitions_full = filtered_homes
demolitions_full.to_csv('../data-processed/demolitions_full.csv')
demolitions_full.shape
```

Out[16]:

```
(3160, 67)
```

Filters specific to partial demolitions

In [17]:

```
# filter all demos to those we designated as Partials
filtered_demotype_partial = demolitions_cut[demolitions_cut['DemoType'] == 'Pa
rtial']
filtered_demotype_partial.shape
```

Out[17]:

```
(5724, 67)
```

In [18]:

```
# look at the WorkClass so we can filter on them.
filtered_demotype_partial.WorkClass.value_counts()
```

Out[18]:

```
Addition and Remodel    4386
Addition                 678
Remodel                  532
New                       72
Repair                    52
Life Safety                4
Name: WorkClass, dtype: int64
```

In [19]:

```
# Filter on WorkClass to get to possible additions
partials_workclass = filtered_demotype_partial[
    (filtered_demotype_partial['WorkClass'] == "Addition and Remodel")
    | (filtered_demotype_partial['WorkClass'] == "Addition")
]
```

In [20]:

```
# Look at PermitClass to filter them
partials_workclass.PermitClass.value_counts()
```

Out[20]:

```
R- 434 Addition & Alterations             5059
R- 438 Residential Garage/Carport Addn       4
R- 436 Addn to increase housing units        1
Name: PermitClass, dtype: int64
```

In [21]:

```
# filter to remove garage/carports
partials_permitclass = partials_workclass[
    (partials_workclass['PermitClass'] != 'R- 438 Residential Garage/Carport A
ddn')
]
partials_permitclass.shape
```

Out[21]:

```
(5060, 67)
```

Set partials dataframe and export

In [22]:

```
demolitions_partial = partials_permitclass
demolitions_partial.to_csv('../data-processed/demolitions_partial.csv')
demolitions_partial.shape
```

Out[22]:

```
(5060, 67)
```

Set demolitions_all dataframe and export

This combines the full and partial demolitions, after filtering, into a single file, in case that is needed.

In [23]:

```
# stack/concat two filtered files
demolitions_all = pd.concat([demolitions_full,demolitions_partial])
demolitions_all.to_csv('../data-processed/demolitions_all.csv')
demolitions_all.shape
```

Out[23]:

```
(8220, 67)
```

Reference

Ignore this. It's just for reference.

```
In [24]: data_typed.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 14277 entries, 0 to 6346
Data columns (total 67 columns):
PermitType                    14277 non-null object
PermitTypeDesc                14277 non-null object
PermitNum                     14277 non-null object
PermitClassMapped             14277 non-null object
PermitClass                   14277 non-null object
WorkClass                     14277 non-null object
Condominium                   14277 non-null object
ProjectName                   14277 non-null object
Description                   14277 non-null object
TCAD_ID                       14184 non-null object
PropertyLegalDescription      13290 non-null object
AppliedDate                   14277 non-null object
IssuedDate                    14277 non-null datetime64[ns]
DayIssued                     14277 non-null object
CalendarYearIssued            14277 non-null object
FiscalYearIssued              14277 non-null int64
IssuedInLast30Days            14277 non-null object
IssuanceMethod                14277 non-null object
StatusCurrent                 14277 non-null object
StatusDate                    14277 non-null object
ExpiresDate                   14277 non-null object
CompletedDate                 9933 non-null object
TotalExistingBldgSQFT         7927 non-null float64
RemodelRepairSQFT             1162 non-null float64
TotalNewAddSQFT               5670 non-null float64
TotalValuationRemodel         5403 non-null float64
TotalJobValuation             13661 non-null float64
NumberOfFloors                14276 non-null float64
HousingUnits                  14276 non-null float64
BuildingValuation             6 non-null float64
BuildingValuationRemodel      5471 non-null float64
ElectricalValuation           6 non-null float64
ElectricalValuationRemodel    5470 non-null float64
MechanicalValuation           6 non-null float64
MechanicalValuationRemodel    5470 non-null float64
PlumbingValuation             6 non-null float64
PlumbingValuationRemodel      5469 non-null float64
MedGasValuation               0 non-null float64
MedGasValuationRemodel        8 non-null float64
OriginalAddress1              14277 non-null object
OriginalCity                  14277 non-null object
OriginalState                 14277 non-null object
OriginalZip                   14277 non-null int64
CouncilDistrict               14224 non-null float64
Jurisdiction                  14277 non-null object
Link                          14277 non-null object
ProjectID                     14277 non-null int64
MasterPermitNum               14222 non-null float64
Latitude                      14277 non-null float64
Longitude                     14277 non-null float64
Location                      14277 non-null object
ContractorTrade               14175 non-null object
ContractorCompanyName         11918 non-null object
ContractorFullName            9323 non-null object
```

```
ContractorPhone              14098 non-null object
ContractorAddress1           10353 non-null object
ContractorAddress2           12965 non-null object
ContractorCity               14146 non-null object
ContractorZip                13756 non-null object
ApplicantFullName            8653 non-null object
ApplicantOrganization        11210 non-null object
ApplicantPhone               13169 non-null object
ApplicantAddress1            9580 non-null object
ApplicantAddress2            12127 non-null object
ApplicantCity                13218 non-null object
ApplicantZip                 12864 non-null object
DemoType                     14277 non-null object
dtypes: datetime64[ns](1), float64(21), int64(3), object(42)
memory usage: 7.4+ MB
```

In [ ]: