Possible partial demolitions

This analysis is based on files downloaded and filtered in the `01_Download_process` notebook in this repo. It is construction permits for "Addition" or "Remodel or Addition" that have the term "partial demo" in the description, definitions we've learned from interviews with City of Austin officials. They are only residential permits for single-family or duplex homes.

It's *IMPORTANT* to note that not all of these permits are demolitions that lead to an increase in the size or shape of the home. In some cases, this might be simply a bathroom or kitchen remodel.

We did a lot of investigation into trying to find "single-wall teardowns", defined as homes where all but one wall was torn down, typically to preserve a historical setback zoning rule. There was not way we could find to find permits for that type of demolition vs any other, with any accuracy.

In [1]:

```python
import pandas as pd
import altair as alt
pd.options.display.max_colwidth = 60
```

In [2]:

```python
# Column type fixes
column_types = {
    'ApplicantPhone': pd.np.str,
    'ContractorPhone': pd.np.str,
    'CalendarYearIssued': pd.np.str,
    'OriginalZip': pd.np.str,
#     'IssuedDate': pd.np.str,
}

# import raw data
data_raw = pd.read_csv(
    '../data-processed/demolitions_partial.csv',
    index_col=None,
    dtype=column_types,
    parse_dates=['IssuedDate']
)

demolitions = data_raw
demolitions.shape
```

Out[2]:

```
(5060, 68)
```

Total cases

In [3]:

```python
print("Partial demolition permits that _could be_ additions: {}".format(len(de
molitions)))
```

```
Partial demolition permits that _could be_ additions: 5060
```

In [4]:

```
# create data from of permits by year
demos_by_year = demolitions.CalendarYearIssued.value_counts().reset_index()
demos_by_year.columns = ['Year', 'Count']
demos_by_year.sort_values('Year')
```

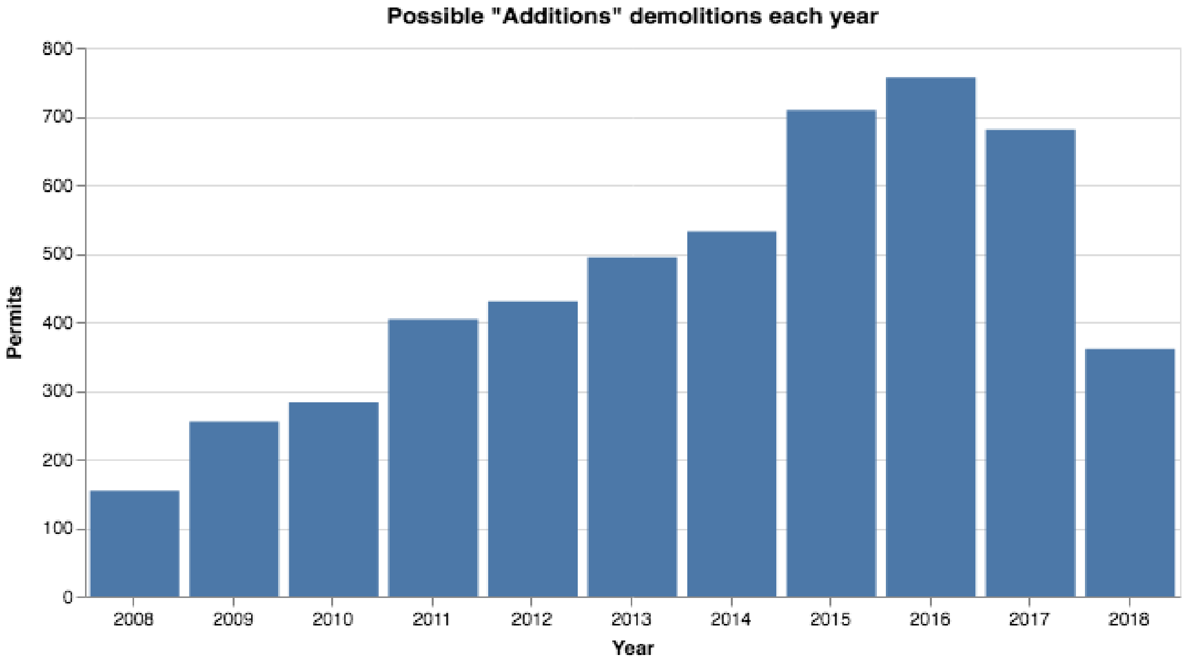
Out[4]:

	Year	Count
10	2008	154
9	2009	255
8	2010	283
6	2011	404
5	2012	430
4	2013	494
3	2014	532
1	2015	709
0	2016	757
2	2017	681
7	2018	361

In [5]:

```
# make chart based on dataframe above
alt.Chart(
    demos_by_year.reset_index(),
    title='Possible "Additions" demolitions each year'
).mark_bar().encode(
    x=alt.X("Year:O", axis=alt.Axis(title="Year", labelAngle=0)),
    y=alt.Y("Count:Q", axis=alt.Axis(title="Permits")),
).properties(width=600)
```

Out[5]:



Partial demos by zip code

In [6]:

```
# make a datafram with counts of permits by zip code
demos_by_zip = demolitions.OriginalZip.value_counts().reset_index()
demos_by_zip.columns = ['Zip', 'Count']
demos_by_zip.head()
```

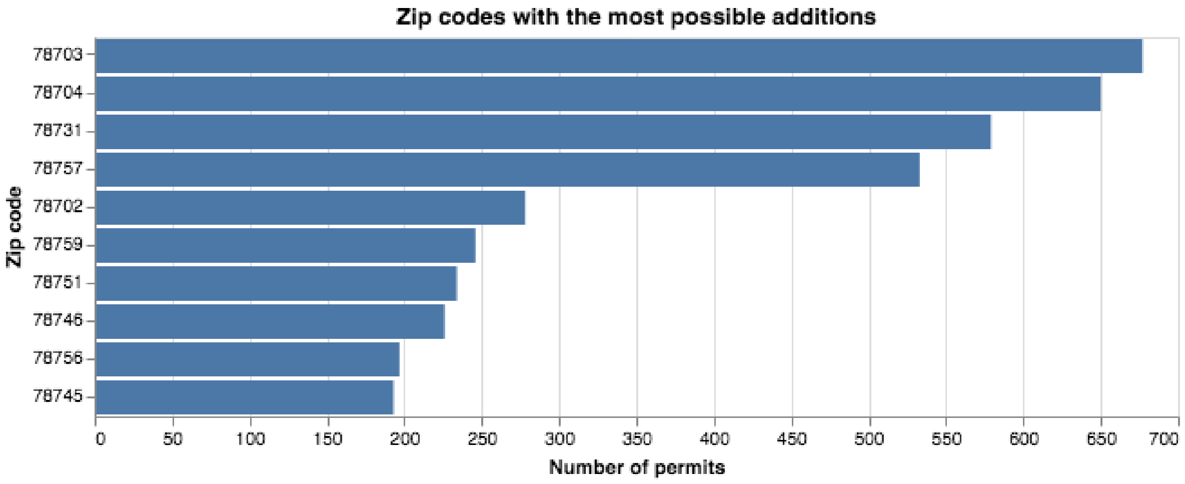
Out[6]:

	Zip	Count
0	78703	677
1	78704	650
2	78731	579
3	78757	533
4	78702	278

In [7]:

```
# chart based on dataframe above
alt.Chart(
    demos_by_zip.head(10),
    title="Zip codes with the most possible additions"
).mark_bar().encode(
    x=alt.X("Count:Q",
            axis=alt.Axis(title="Number of permits")),
    y=alt.Y("Zip:O",
            # this is the sorting magic
            sort=alt.SortField(
                field="Count",
                order="descending",
                op="sum"
            ),
            axis=alt.Axis(title="Zip code")),
).properties(width=600)
```

Out[7]:



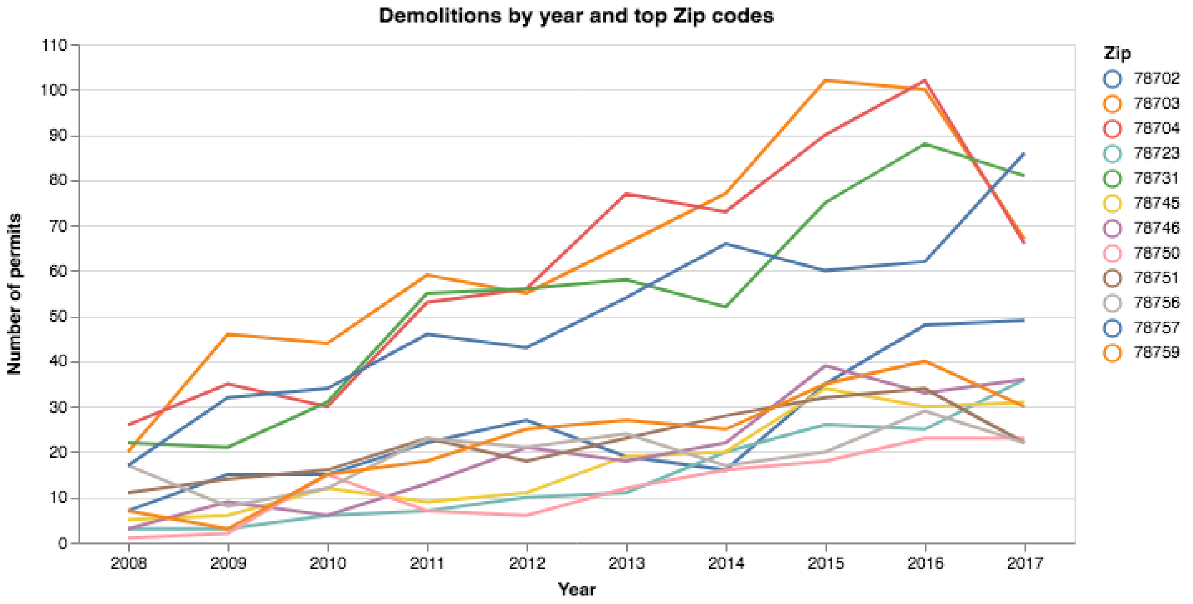
Possible additions by zip by year

This chart is interesting in that it shows the hottest zip codes (03 and 04) had drops in 2017, falling out of the lede for the first time since 2012. Not sure it is worth finding out why at this point, but... it's interesting.

In [8]:

```
# Filter out 2018 because it is a partial year
before_2018 = demolitions[demolitions['CalendarYearIssued'] != '2018']

# build a dataframe of yr, zip, count
demos_yr_zip = before_2018.groupby(['CalendarYearIssued','OriginalZip']).agg(dict(
    PermitNum='count'
)).reset_index()
demos_yr_zip.columns = ['Year', 'Zip', 'Count']
demos_yr_zip.shape
# demos_yr_zip.head()

# filter to 2017
zips_2017 = demos_yr_zip[demos_yr_zip['Year'] == '2017']

# filter the 2017 list to those with 20+ demos
# sort the list by count descending
zips_2017_top = zips_2017[zips_2017.Count > 19].sort_values('Count', ascending=False)

# Use the Zip code column above to filter the original demos_yr_zip datatable
# and create the new dataframe to chart
top_demos_yr_zip_new = demos_yr_zip[demos_yr_zip['Zip'].isin(zips_2017_top.Zip)]

alt.Chart(
    top_demos_yr_zip_new,
    title="Demolitions by year and top Zip codes"
).mark_line().encode(
    x=alt.X("Year:O",
            axis=alt.Axis(title="Year", labelAngle=0)),
    y=alt.Y("Count:Q",
            axis=alt.Axis(title="Number of permits")),
    color="Zip"
).properties(width=600)
```

Out[8]:



In [ ]: