

```
In [12]:  import pandas as pd
```

```
In [13]:  wzfatal = pd.read_csv('../data/Work_Zone_Fatal_Crashes_Fatalities_Data.csv')
wzfatal.head(1000)
```

...

```
In [14]:  wzfatalTN = pd.read_csv('../data/Work_Zone_Fatal_Crashes_Fatalities_Data_TN.csv')
```

```
In [15]:  wzfatalTN.head(100)
```

...

```
In [16]:  wzfatalTN2 = wzfatalTN[wzfatalTN['State'].str.contains('Tennessee')]
```

```
In [17]:  wzfatalTN2
```

...

```
In [18]:  wzfatalTN3 = wzfatalTN2.sort_values(by='Year')
```

```
In [19]:  wzfatalTN3
```

Out[19]:

	State	Year	Work Zone Fatalities	Work Zone Fatal Crashes	Work Zone Truck- Involved Fatal Crashes	Work Zone Truck- Involved Fatalities	Work Zone Bus- Involved Fatal Crashes	Work Zone Bus- Involved Fatalities	Work Zone Pedestrian- Involved Fatal Crashes
343	Tennessee	2009	15.0	13.0	3.0	4.0	0.0	0.0	2.0
342	Tennessee	2010	7.0	7.0	0.0	0.0	1.0	1.0	1.0
341	Tennessee	2011	16.0	16.0	5.0	5.0	0.0	0.0	4.0
340	Tennessee	2012	12.0	11.0	5.0	6.0	0.0	0.0	2.0
339	Tennessee	2013	14.0	13.0	8.0	9.0	0.0	0.0	3.0
338	Tennessee	2014	21.0	20.0	5.0	5.0	0.0	0.0	5.0
337	Tennessee	2015	17.0	14.0	4.0	4.0	0.0	0.0	4.0
336	Tennessee	2016	13.0	12.0	2.0	2.0	0.0	0.0	2.0
466	Tennessee	2017	10.0	10.0	1.0	1.0	0.0	0.0	3.0
519	Tennessee	2018	20.0	19.0	7.0	7.0	0.0	0.0	3.0



```
In [23]:  census = pd.read_csv('../data/QuickFacts Feb-26-2020.csv')
census
```

...

In [24]: `census.head(10)`

Out[24]:

	Fact	Fact Note	Davidson County, Tennessee	Value Note for Davidson County, Tennessee
0	Population estimates, July 1, 2019, (V2019)	NaN	NaN	NaN
1	Population estimates, July 1, 2018, (V2018)	NaN	692,587	NaN
2	Population estimates base, April 1, 2010, (V2010)	NaN	NaN	NaN
3	Population estimates base, April 1, 2010, (V2010)	NaN	626,560	NaN
4	Population, percent change - April 1, 2010 (estimated)	NaN	NaN	NaN
5	Population, percent change - April 1, 2010 (estimated)	NaN	10.5%	NaN
6	Population, Census, April 1, 2010	NaN	626,681	NaN
7	Persons under 5 years, percent	NaN	6.6%	NaN
8	Persons under 18 years, percent	NaN	20.8%	NaN
9	Persons 65 years and over, percent	NaN	12.2%	NaN

In [25]: `census2 = pd.DataFrame([[2010, 2018, 2030, 2040], [627, 693, None, 850]],
columns=['year', 'pop'],
census2`

...

In [26]: `import numpy as np

simple array
data = np.array(['627', '693', 'None', '850'])

providing an index
ser = pd.Series(data, index=[2010, 2018, 2030, 2040])
print(ser)`

```
2010    627
2018    693
2030    None
2040    850
dtype: object
```

```
In [ ]: 8. store month from date entered column in building permits dataframe to a new column
building_permits['month_entered'] = building_permits['date_entered'].str.split('-', expand=True)[0]
ask what did John do for this one?
```

Find the value counts of month entered, then reset the index to convert a pd series to a df

```
month_count = building_permits['month_entered'].value_counts().reset_index()
```

rename the columns to better reflect ..?

```
month_count.columns = ['Month_of_Year', 'Permit_Counts']
```

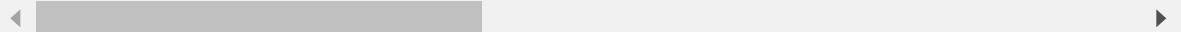
9. Make a bar plot from month count using matplotlib that shows the number of permits issued by month

sort the month of year column using default (ascending = True)

```
month_count = month_count.sort_values('Month_of_Year')
```

create a bar plot from our month count df with month of year on the x axis and permit counts on the y axis

```
ax = month_count.plot.bar(x='Month_of_Year', y='Permit_Counts')
```

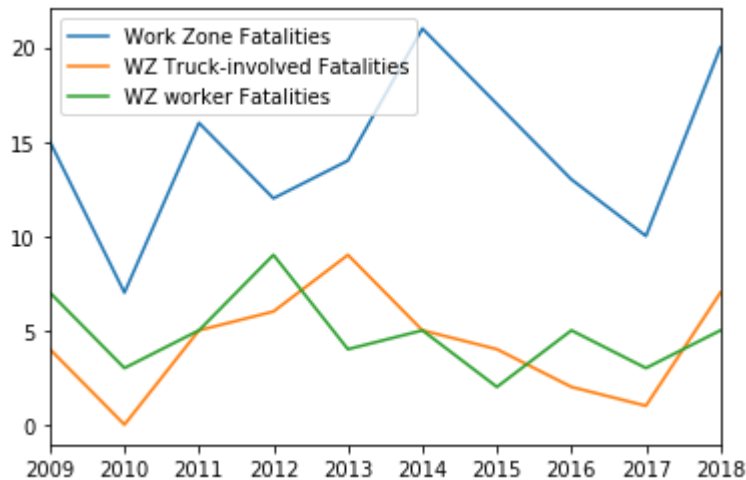


```
In [42]: fatalities_year = wzfatalTN3['Year'].reset_index()
```

```
In [48]: wzfatalTN3.columns = ['Year', 'Work Zone Fatalities']
```

...

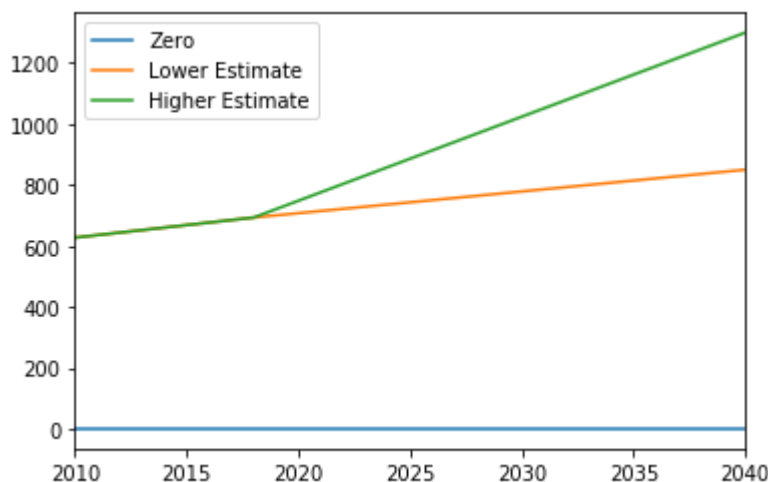
```
In [60]: WZF = pd.DataFrame({
    'Work Zone Fatalities': [15, 7, 16, 12, 14, 21, 17, 13, 10, 20],
    'WZ Truck-involved Fatalities': [4, 0, 5, 6, 9, 5, 4, 2, 1, 7],
    'WZ worker Fatalities': [7, 3, 5, 9, 4, 5, 2, 5, 3, 5]
    }, index=[2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018])
WZFlines = WZF.plot.line()
```



```
data = np.array(['627', '693', '850'])
```

```
ser = pd.Series(data, index=[2010, 2018, 2040]) print(ser)
```

```
In [68]: county_growth = pd.DataFrame({
    'Zero': [0, 0, 0],
    'Lower Estimate': [627, 693, 850],
    'Higher Estimate': [627, 693, 1300],
    }, index=[2010, 2018, 2040])
county_growthlines = county_growth.plot.line()
```



```
In [ ]: NEAR 28th N at Jefferson st
Davidson County, CENTENNIAL BLVD - NEAR I-40
Route: 0A999

AADT by year:
2018 = 3,489
2017 = 3,688
2016 = 3,651
2015 = 3,615
2014 = 9,031
2013 = 3,963
2012 = 6,850
2011 = 3,820
2010 = 6,941
2009 = 10,217
2008 = 10,415

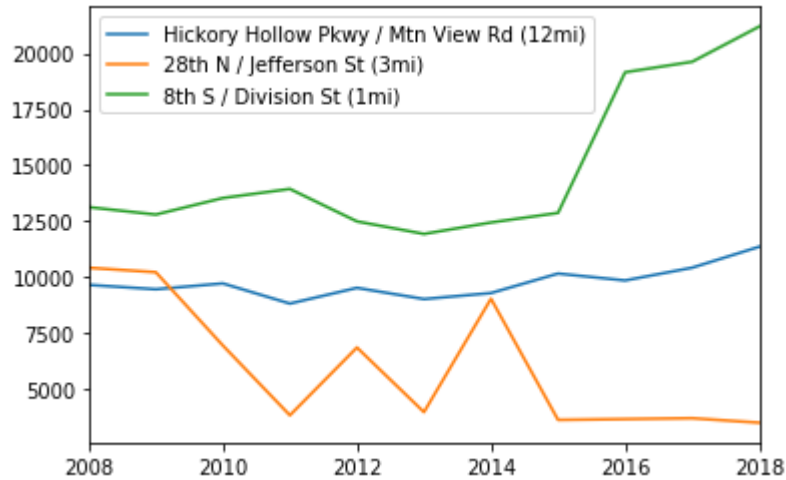
NEAR 8th and Division
Davidson County, 8TH AVE S-B/T GLEAVERS & FOGG
Route: SR006

AADT by year:
2018 = 21,200
2017 = 19,626
2016 = 19,156
2015 = 12,868
2014 = 12,440
2013 = 11,928
2012 = 12,489
2011 = 13,940
2010 = 13,534
2009 = 12,794
2008 = 13,126

NEAR hickory hollow pkwy at mtn view rd
Davidson County, ANTIOCH PK - S OF HAYWOOD LN
Route: 04169

AADT by year:
2018 = 11,355
2017 = 10,418
2016 = 9,848
2015 = 10,156
2014 = 9,285
2013 = 9,016
2012 = 9,517
2011 = 8,814
2010 = 9,715
2009 = 9,455
2008 = 9,650
```

```
In [70]: ▶ intersect = pd.DataFrame({  
    'Hickory Hollow Pkwy / Mtn View Rd (12mi)': [9650, 9455, 9715, 8814, 9517,  
    '28th N / Jefferson St (3mi)': [10415, 10217, 6941, 3820, 6850, 3963, 9031,  
    '8th S / Division St (1mi)': [13126, 12794, 13534, 13940, 12489, 11928, 12  
    }, index=[2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018  
    intersectlines = intersect.plot.line()
```



In []: ▶

24S

Davidson County, S OF BRILEY PKWY (ATR 33)

Route: I0024

AADT by year:

2018 = 190,529

2017 = 186,329

2016 = 176,880

2015 = 169,896

2014 = 172,117

2013 = 170,275

2012 = 169,431

2011 = 164,092

2010 = 150,736

2009 = 155,914

2008 = 155,369

SE INNER LOOP

Davidson County, S OF SILLMAN EVANS BRIDGE

Route: I0040

AADT by year:

2018 = 174,018

2017 = 153,345

2016 = 169,662

2015 = 180,528

2014 = 169,085

2013 = 168,310

2012 = 170,005

2011 = 159,804

2010 = 156,560

2009 = 153,568

2008 = 152,367

EAST NASHVILLE 65N

Davidson County, [LOOPS] S OF TRINITY LN

Route: I0065

AADT by year:

2018 = 183,350

2017 = 176,107

2016 = 158,423

2015 = 169,301

2014 = 164,671

2013 = 166,924

2012 = 161,637

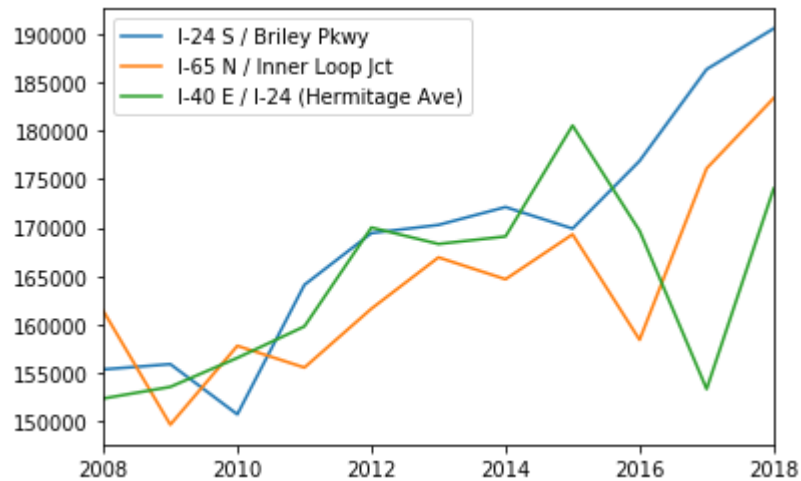
2011 = 155,561

2010 = 157,801

2009 = 149,668

2008 = 161,410

```
In [72]: ▶ hotspots = pd.DataFrame({  
    'I-24 S / Briley Pkwy': [155369, 155914, 150736, 164092, 169431, 170275, 1  
    'I-65 N / Inner Loop Jct': [161410, 149668, 157801, 155561, 161637, 166924  
    'I-40 E / I-24 (Hermitage Ave)': [152367, 153568, 156560, 159804, 170005,  
    }, index=[2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018  
    hotspotslines = hotspots.plot.line()
```



```
In [ ]: ▶
```