In [2]:
```python
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

data_filename = 'shootings.csv'
df = pd.read_csv(data_filename)
```

In [3]:
```python
df.describe
#ask Questions
#Which race had the highest mortality rate
#Biggest factor on the desicion to kill
#which city had the highest deaths
# does gender play a role?
#what date has the highest mortality rate
```

Out[3]:
```
<bound method NDFrame.describe of         id                name        date   manner_of_death       armed  \
0        3          Tim Elliot  2015-01-02              shot         gun
1        4    Lewis Lee Lembke  2015-01-02              shot         gun
2        5  John Paul Quintero  2015-01-03  shot and Tasered     unarmed
3        8     Matthew Hoffman  2015-01-04              shot  toy weapon
4        9   Michael Rodriguez  2015-01-04              shot    nail gun
...    ...                 ...         ...               ...         ...
4890  5916     Rayshard Brooks  2020-06-12              shot       Taser
4891  5925      Caine Van Pelt  2020-06-12              shot         gun
4892  5918        Hannah Fizer  2020-06-13              shot     unarmed
4893  5921      William Slyter  2020-06-13              shot         gun
4894  5924      Nicholas Hirsh  2020-06-15              shot         gun

       age gender      race           city state  signs_of_mental_illness  \
0     53.0      M     Asian        Shelton    WA                     True
1     47.0      M     White          Aloha    OR                    False
2     23.0      M  Hispanic        Wichita    KS                    False
3     32.0      M     White  San Francisco    CA                     True
4     39.0      M  Hispanic          Evans    CO                    False
...    ...    ...       ...            ...   ...                      ...
4890  27.0      M     Black        Atlanta    GA                    False
4891  23.0      M     Black    Crown Point    IN                    False
4892  25.0      F     White        Sedalia    MO                    False
4893  22.0      M     White    Kansas City    MO                    False
4894  31.0      M     White       Lawrence    KS                    False

     threat_level         flee  body_camera          arms_category
0          attack  Not fleeing        False                   Guns
1          attack  Not fleeing        False                   Guns
2           other  Not fleeing        False                Unarmed
3          attack  Not fleeing        False  Other unusual objects
4          attack  Not fleeing        False       Piercing objects
...           ...          ...          ...                    ...
4890       attack         Foot         True     Electrical devices
4891       attack          Car        False                   Guns
4892        other  Not fleeing        False                Unarmed
4893        other        Other        False                   Guns
4894       attack          Car        False                   Guns

[4895 rows x 15 columns]>
```

In [4]:
```python
df.columns
```

Out[4]:
```
Index(['id', 'name', 'date', 'manner_of_death', 'armed', 'age', 'gender',
       'race', 'city', 'state', 'signs_of_mental_illness', 'threat_level',
       'flee', 'body_camera', 'arms_category'],
      dtype='object')
```

In [5]:
```python
df.manner_of_death.unique()
```

Out[5]:
```
array(['shot', 'shot and Tasered'], dtype=object)
```

In [6]:
```python
Unique_Description = df.manner_of_death.unique()
Percentage_Description_count = (df.manner_of_death.value_counts() / len(df))*100
Percentage_Description_count
```

Out[6]:
```
shot                94.933606
shot and Tasered     5.066394
Name: manner_of_death, dtype: float64
```

In [ ]:

In [7]:
```python
Unique_Description2 = df.armed.unique()
Percentage_Description_count2 = (df.armed.value_counts() / len(df))*100
Percentage_Description_count2
```

Out[7]:
```
gun             56.281920
knife           14.463739
unknown          8.539326
unarmed          7.109295
toy weapon       3.493361
                  ...
beer bottle      0.020429
fireworks        0.020429
flagpole         0.020429
pepper spray     0.020429
stapler          0.020429
Name: armed, Length: 89, dtype: float64
```

In [8]:
```python
Percentage_Description_count3 = (df.age.value_counts() / len(df))*100
Percentage_Description_count3
```

Out[8]:
```
25.0    3.595506
31.0    3.554648
27.0    3.370787
33.0    3.350358
29.0    3.268641
          ...
12.0    0.020429
13.0    0.020429
78.0    0.020429
79.0    0.020429
81.0    0.020429
Name: age, Length: 76, dtype: float64
```

In [9]:
```python
Unique_Description4 = df.gender.unique()
Unique_Description4
Percentage_Description_count4 = (df.gender.value_counts() / len(df))*100
Percentage_Description_count4
```

Out[9]:
```
M    95.46476
F     4.53524
Name: gender, dtype: float64
```

In [10]:
```python
Unique_Description5 = df.race.unique()
Unique_Description5
Percentage_Description_count5 = (df.race.value_counts() / len(df))*100
Percentage_Description_count5
```

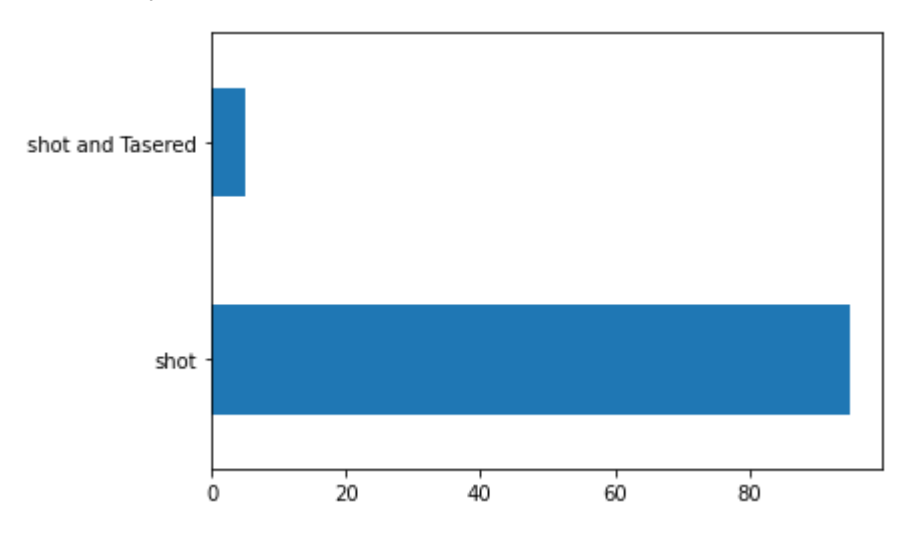
Out[10]:
```
White       50.582227
Black       26.516854
Hispanic    18.426966
Asian        1.899898
Native       1.593463
Other        0.980592
Name: race, dtype: float64
```

In [11]:
```python
Unique_Description6 = df.city.unique()
Percentage_Description_count6 = (df.city.value_counts() / len(df))*100
Percentage_Description_count6
```

Out[11]:
```
Los Angeles     1.593463
Phoenix         1.348315
Houston         1.041879
Las Vegas       0.837589
San Antonio     0.817160
                  ...
Escalon         0.020429
Kasilof         0.020429
Caddo Parish    0.020429
Galveston       0.020429
Markham         0.020429
Name: city, Length: 2288, dtype: float64
```

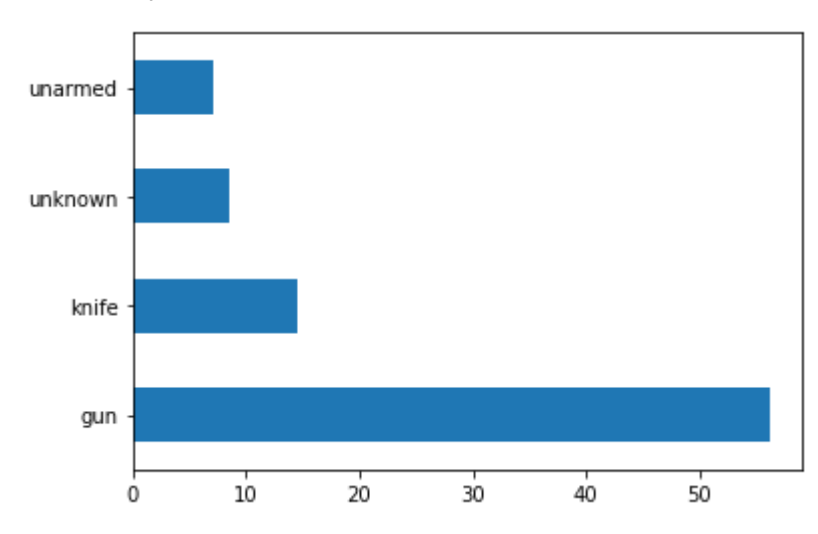
In [12]:
```python
Percentage_Description_count.plot(kind='barh')
```

Out[12]: <AxesSubplot:>



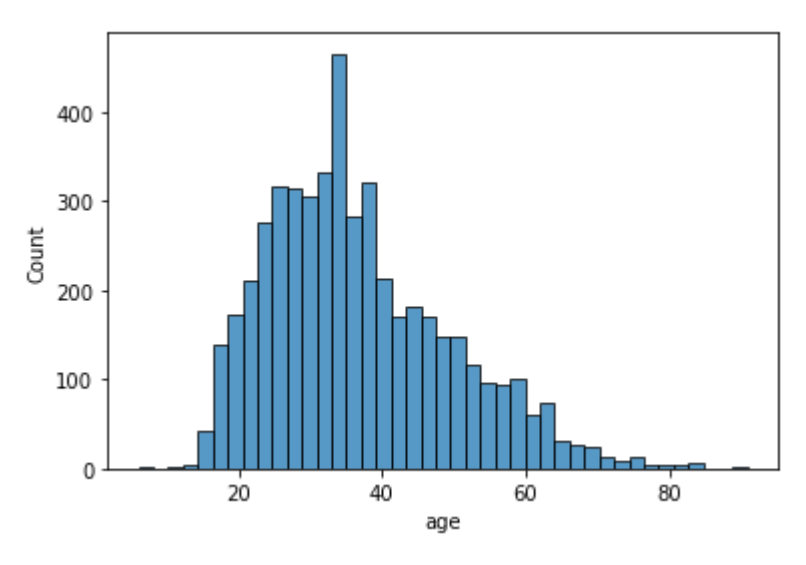
In [13]:
```python
Percentage_Description_count2[Percentage_Description_count2 > 5].plot(kind='barh')
```

Out[13]: <AxesSubplot:>



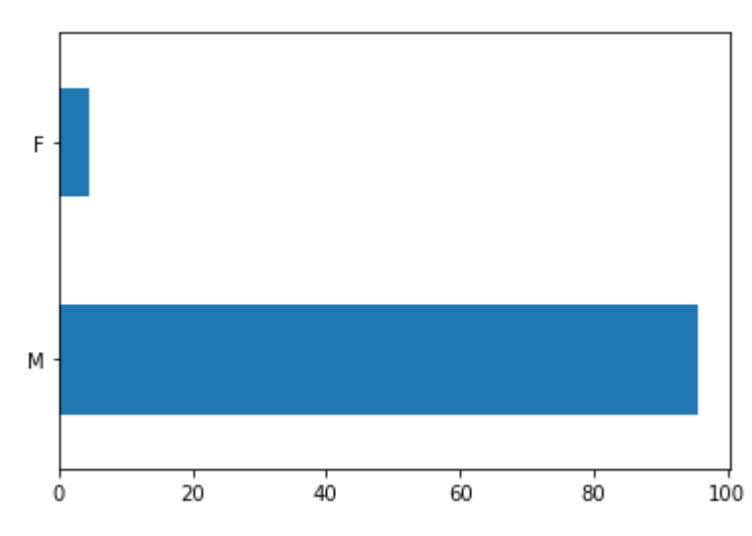
In [35]:
```python
sns.histplot(df.age)
```

Out[35]: <AxesSubplot:xlabel='age', ylabel='Count'>



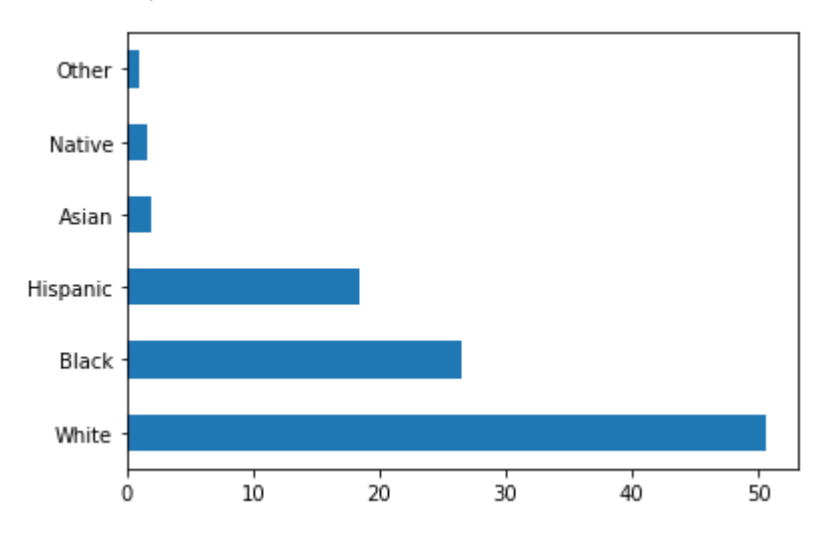
In [15]:
```python
Percentage_Description_count4[Percentage_Description_count4 !=0].plot(kind='barh')
```

Out[15]: <AxesSubplot:>



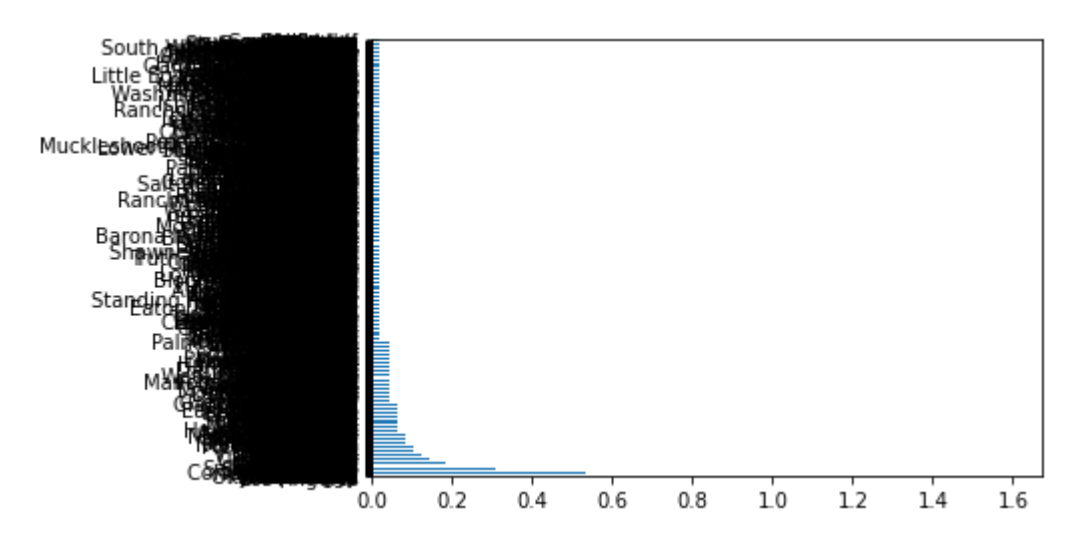
In [16]:
```python
Percentage_Description_count5[Percentage_Description_count5 !=0].plot(kind='barh')
```

Out[16]: <AxesSubplot:>



In [17]:
```python
Percentage_Description_count6[Percentage_Description_count6 >= 0].plot(kind='barh')
```

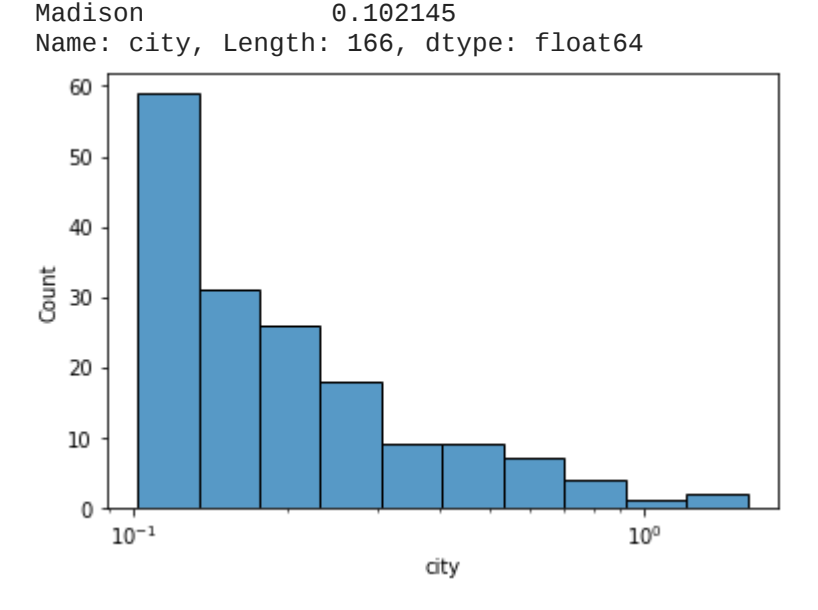
Out[17]: <AxesSubplot:>



In [18]:
```python
#since cities are so widely dispraportionate its better to have two graphs each showcasing the extreme of both ends of the graph
high_death_cities= Percentage_Description_count6[Percentage_Description_count6 >= 0.1]
low_death_cities= Percentage_Description_count6[Percentage_Description_count6 < 0.1]
```

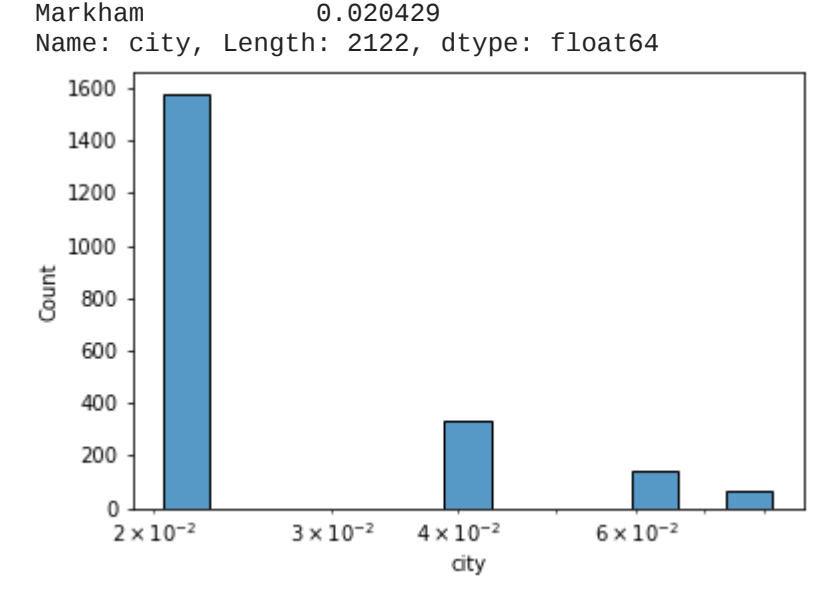
In [19]:
```python
sns.histplot(high_death_cities,log_scale=True)
high_death_cities
#From exponential decrease to increased
```

Out[19]:
```
Los Angeles        1.593463
Phoenix            1.348315
Houston            1.041879
Las Vegas          0.837589
San Antonio        0.817160
                     ...
Elkton             0.102145
Casper             0.102145
Gulfport           0.102145
Fort Lauderdale    0.102145
Madison            0.102145
Name: city, Length: 166, dtype: float64
```



In [20]:
```python
sns.histplot(low_death_cities, log_scale=True)
low_death_cities
#exponentially decreasing
```

Out[20]:
```
Mission           0.081716
Sunnyvale         0.081716
Laredo            0.081716
Paris             0.081716
St. Petersburg    0.081716
                    ...
Escalon           0.020429
Kasilof           0.020429
Caddo Parish      0.020429
Galveston         0.020429
Markham           0.020429
Name: city, Length: 2122, dtype: float64
```



In [21]:
```python
df.date
```

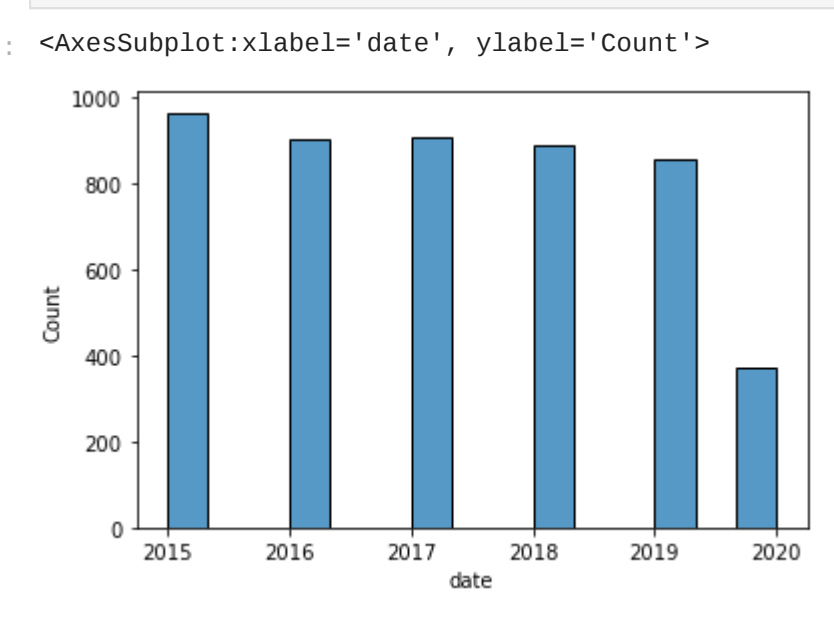
Out[21]:
```
0       2015-01-02
1       2015-01-02
2       2015-01-03
3       2015-01-04
4       2015-01-04
           ...
4890    2020-06-12
4891    2020-06-12
4892    2020-06-13
4893    2020-06-13
4894    2020-06-15
Name: date, Length: 4895, dtype: object
```

In [22]:
```python
#convert date to timestamp to get a more accurate repressentation of the data
```

In [23]:
```python
df.date= pd.to_datetime(df.date)
```

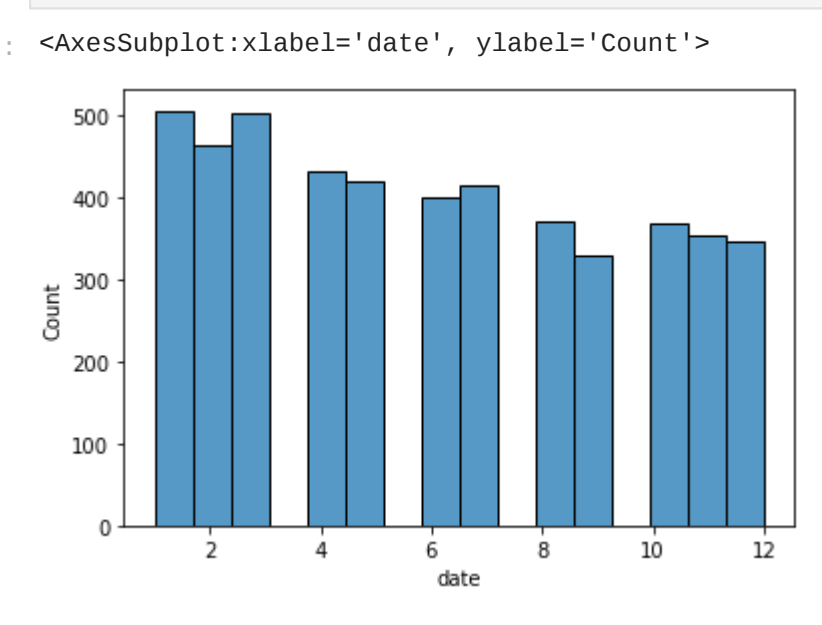
In [24]:
```python
#we will focus on getting year first
sns.histplot(df.date.dt.year)
```

Out[24]: <AxesSubplot:xlabel='date', ylabel='Count'>



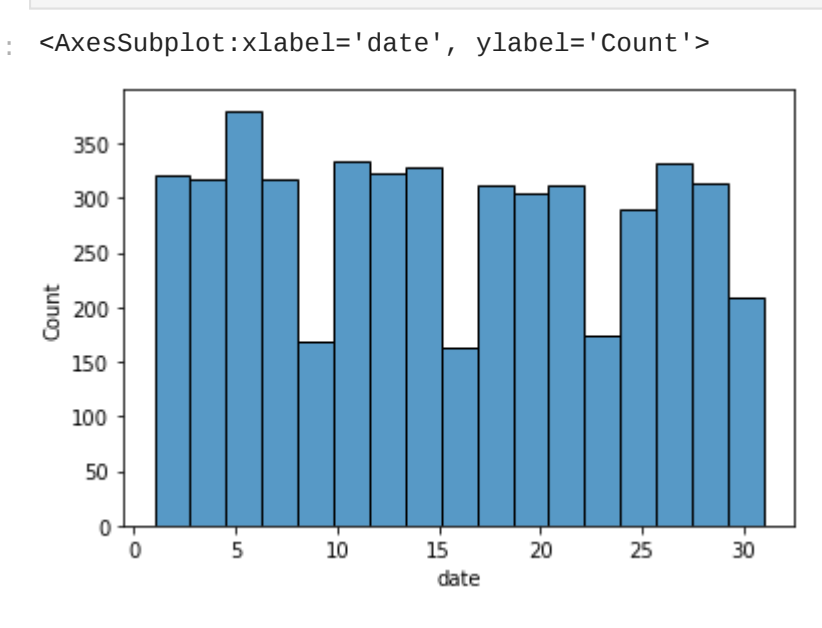
In [25]:
```python
sns.histplot(df.date.dt.month)
```

Out[25]: <AxesSubplot:xlabel='date', ylabel='Count'>



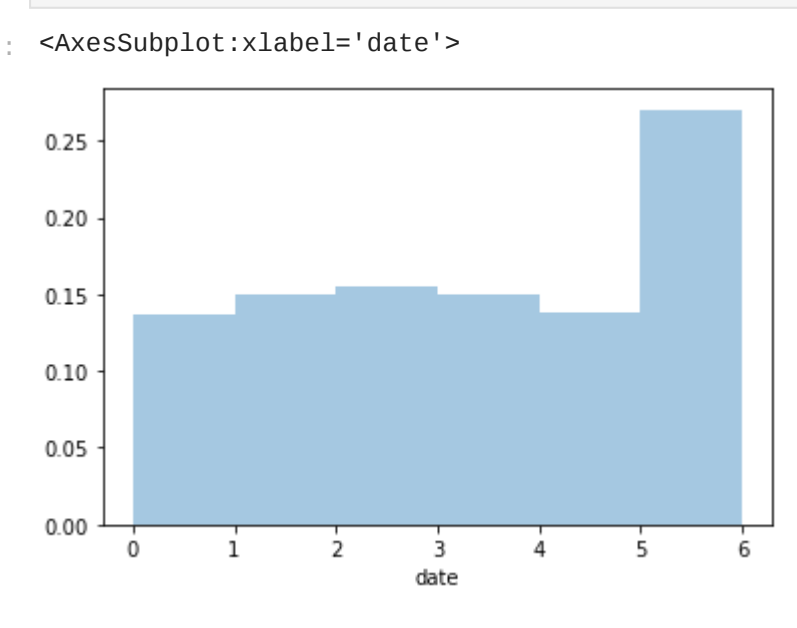
In [26]:
```python
sns.histplot(df.date.dt.day)
```

Out[26]: <AxesSubplot:xlabel='date', ylabel='Count'>



In [46]:
```python
#mosst fatal shootings occur on weekends especially on sunday
sns.distplot(df.date.dt.dayofweek, bins= 6 , kde=False , norm_hist=True)
```

Out[46]: <AxesSubplot:xlabel='date'>



In [ ]:
```python
#the weight of evidence regression
#after analysing the data we can make up a hypothesis in regards to the overall effect of certain varaibles when paired against one another
```

In [47]:
```python
df.corr()
```

Out[47]:

	id	age	signs_of_mental_illness	body_camera
id	1.000000	0.022361	-0.086920	0.030910
age	0.022361	1.000000	0.096865	-0.035463
signs_of_mental_illness	-0.086920	0.096865	1.000000	0.049636
body_camera	0.030910	-0.035463	0.049636	1.000000

In [ ]:

In [ ]: