Animal_Rescues

May 30, 2022

[268]:
```
# importing
import pandas as pd
import geopandas as gpd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import matplotlib.mlab as mlab
import scipy.stats as st

%matplotlib inline
sns.set_style("whitegrid")
plt.style.use("fivethirtyeight")
```

[271]:
```
# reading in the data
url = 'https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/
 ↪data/2021/2021-06-29/animal_rescues.csv'
df = pd.read_csv(url)
df.head()
```

[271]:
```
   incident_number date_time_of_call  cal_year fin_year type_of_incident  \
0         139091.0  01/01/2009 03:01      2009  2008/09  Special Service
1         275091.0  01/01/2009 08:51      2009  2008/09  Special Service
2        2075091.0  04/01/2009 10:07      2009  2008/09  Special Service
3        2872091.0  05/01/2009 12:27      2009  2008/09  Special Service
4        3553091.0  06/01/2009 15:23      2009  2008/09  Special Service

   pump_count  pump_hours_total  hourly_notional_cost  incident_notional_cost  \
0         1.0               2.0                   255                   510.0
1         1.0               1.0                   255                   255.0
2         1.0               1.0                   255                   255.0
3         1.0               1.0                   255                   255.0
4         1.0               1.0                   255                   255.0

  final_description  …          uprn           street        usrn  \
0          Redacted  …           NaN  Waddington Way  20500146.0
1          Redacted  …           NaN   Grasmere Road         NaN
2          Redacted  …           NaN       Mill Lane         NaN
3          Redacted  …  1.000210e+11       Park Lane  21401484.0
```

```
4          Redacted  …           NaN    Swindon Lane  21300122.0

  postcode_district easting_m northing_m easting_rounded northing_rounded  \
0              SE19       NaN        NaN          532350           170050
1              SE25  534785.0   167546.0          534750           167550
2               SM5  528041.0   164923.0          528050           164950
3               UB9  504689.0   190685.0          504650           190650
4               RM3       NaN        NaN          554650           192350

    latitude longitude
0        NaN       NaN
1  51.390954 -0.064167
2  51.368941 -0.161985
3  51.605283 -0.489684
4        NaN       NaN

[5 rows x 31 columns]
```

[204]:

```
# type of incident all same val
cols = ['cal_year', 'fin_year','pump_count', 'pump_hours_total',
       'hourly_notional_cost', 'incident_notional_cost', 'final_description',
       'animal_group_parent', 'property_type',
       'property_category', 'special_service_type_category',
       'special_service_type', 'borough_code', 'borough',
       'latitude', 'longitude']
```

[205]:

```
# looking at df col names
df.columns
df = df[cols]
```

[206]:

```
# looking for nulls in df
df.isnull().sum()
```

[206]:

```
cal_year                          0
fin_year                          0
pump_count                       50
pump_hours_total                 51
hourly_notional_cost              0
incident_notional_cost           51
final_description                 5
animal_group_parent               0
property_type                     0
property_category                 0
special_service_type_category     0
special_service_type              0
borough_code                      9
borough                           9
```

```
latitude                         3843
longitude                        3843
dtype: int64
```

[207]:

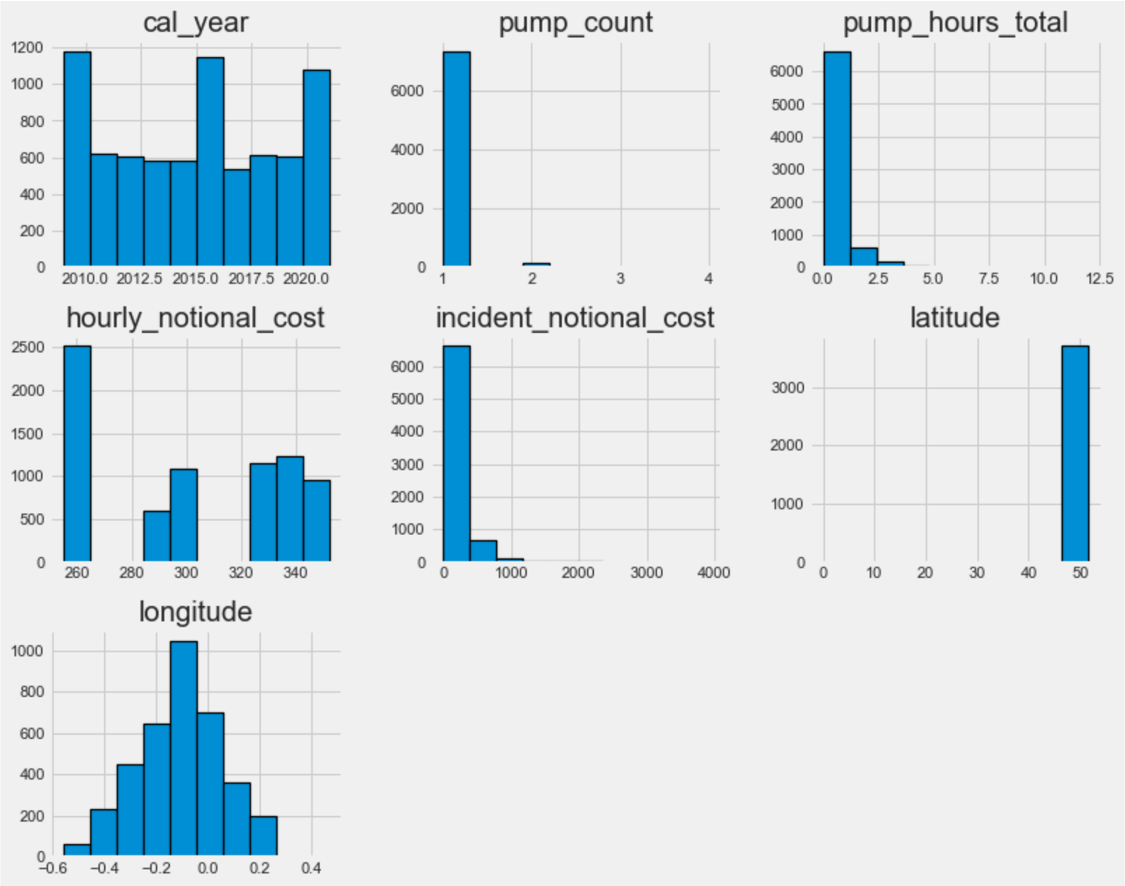
```
# descriptive statistics for each column (doesnt make sense with all the cols)
df.describe()
```

[207]:

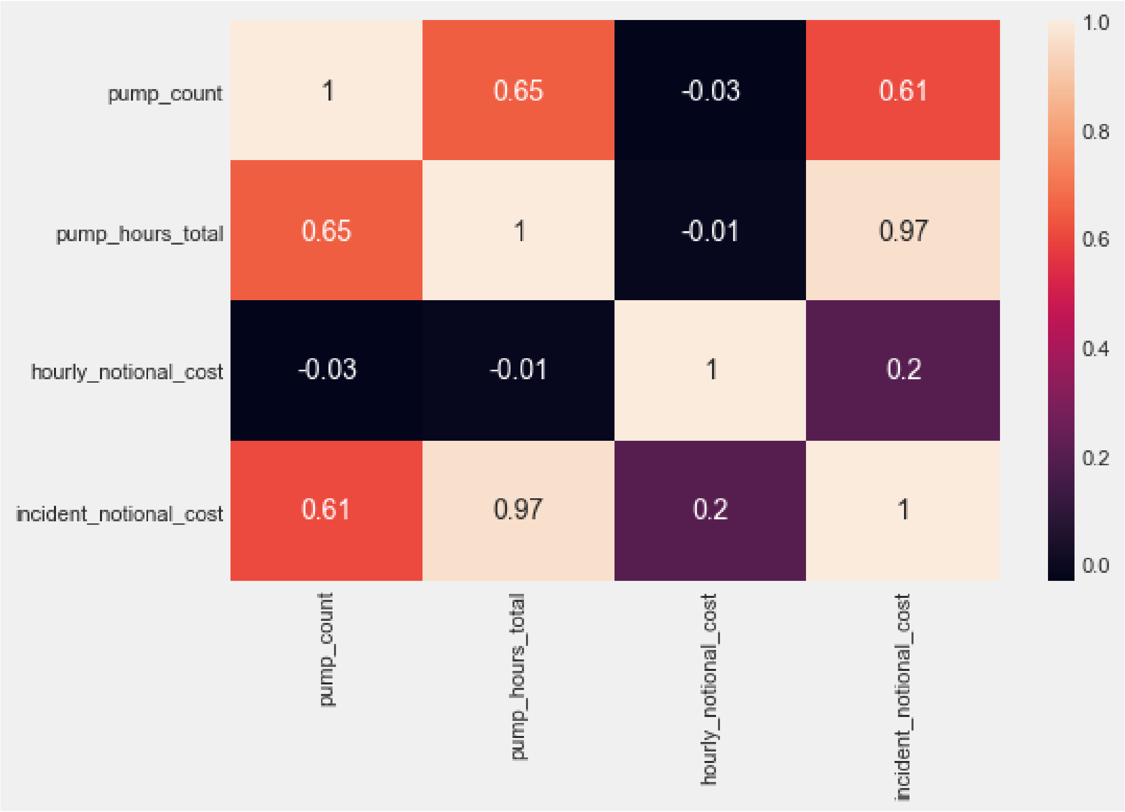
```
          cal_year   pump_count  pump_hours_total  hourly_notional_cost  \
count  7544.000000  7494.000000       7493.000000           7544.000000
mean   2014.884279     1.020283          1.181770            301.262593
std       3.681615     0.152788          0.636624             33.992573
min    2009.000000     1.000000          0.000000            255.000000
25%    2012.000000     1.000000          1.000000            260.000000
50%    2015.000000     1.000000          1.000000            298.000000
75%    2018.000000     1.000000          1.000000            333.000000
max    2021.000000     4.000000         12.000000            352.000000

       incident_notional_cost     latitude    longitude
count             7493.000000  3701.000000  3701.000000
mean               355.764580    51.496734    -0.110938
std                194.646035     0.850323     0.161945
min                  0.000000     0.000000    -0.559730
25%                260.000000    51.456916    -0.215739
50%                326.000000    51.514828    -0.102124
75%                339.000000    51.568187    -0.010953
max               3912.000000    51.688304     0.466421
```

[208]:

```
df.hist(edgecolor='black', linewidth=1.2, figsize=(12, 10))
plt.show()
```



```
[209]: # not all vals worth correlating - can add more or less that are worth it
corr_df = df[['pump_count', 'pump_hours_total',
        'hourly_notional_cost', 'incident_notional_cost']]
corr_matrix = corr_df.corr().round(2)
#heatmap plotting
plt.figure(figsize = (8,5))
sns.heatmap(corr_matrix, annot = True) #Annot = True is used to print values inside the square
plt.show()
```



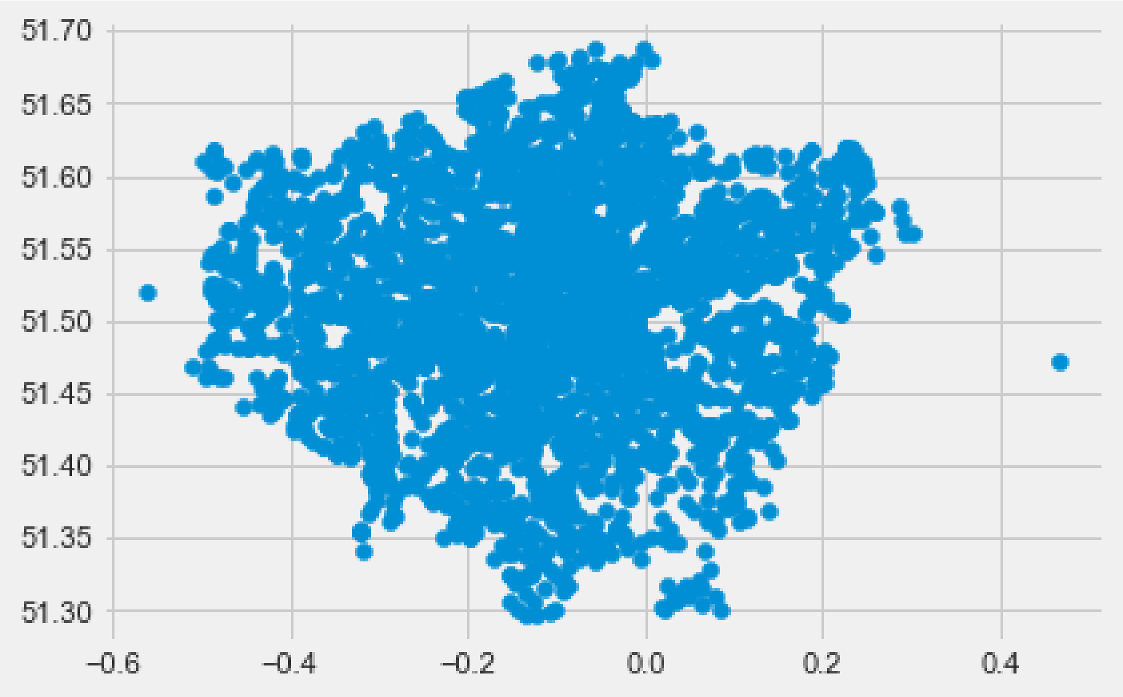
[210]:
```
lat = df[['latitude']].sort_values(by=['latitude']).dropna()
lat.head()
## THERE IS 0 LAT OF 0.0 WHICH IS NOT LONDON THIS NEEDS TO BE DROPPED
```

[210]:
```
       latitude
7028   0.000000
7145  51.297695
4652  51.297698
2102  51.298263
4134  51.300551
```

[211]:
```
# dropped by index ## worked
cdf = df.drop(7028)
latt = cdf[['latitude']].sort_values(by=['latitude']).dropna()
latt.head()
```

[211]:
```
       latitude
7145  51.297695
4652  51.297698
2102  51.298263
4134  51.300551
4709  51.300713
```

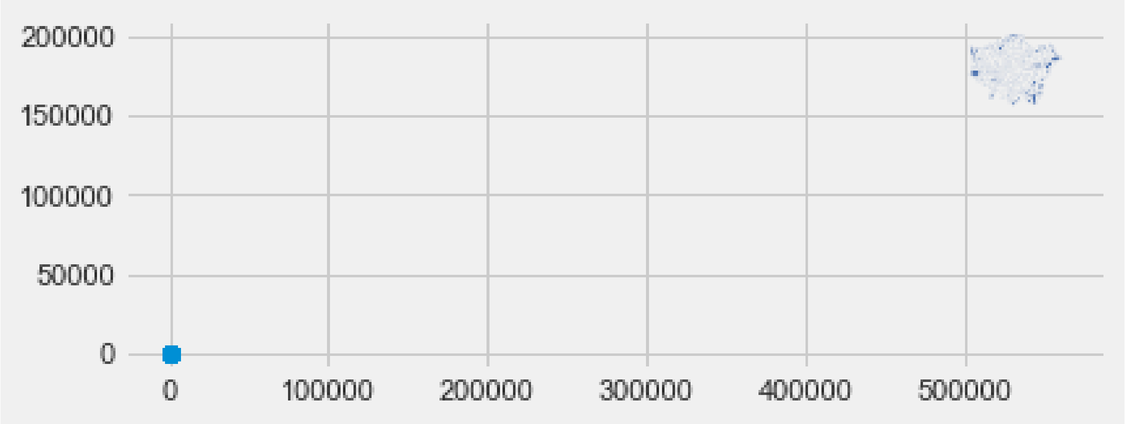
[212]:
```
#points no london background
plt.scatter(x=cdf['longitude'], y=cdf['latitude'])
plt.show()
```



[213]:
```
#plotting with base map of london shape file from
# https://towardsdatascience.com/
 ↪lets-make-a-map-using-geopandas-pandas-and-matplotlib-to-make-a-chloropleth-map-dddc31c1983
#need to pip install gpd

map_df = gpd.read_file('London_Ward.shp')
map_df.head()
```

[213]:
```
                     NAME   GSS_CODE             DISTRICT  LAGSSCODE  \
0       Chessington South  E05000405  Kingston upon Thames  E09000021
1  Tolworth and Hook Rise  E05000414  Kingston upon Thames  E09000021
2              Berrylands  E05000401  Kingston upon Thames  E09000021
3               Alexandra  E05000400  Kingston upon Thames  E09000021
4                Beverley  E05000402  Kingston upon Thames  E09000021

   HECTARES  NONLD_AREA                                           geometry
0   755.173         0.0  POLYGON ((516401.596 160201.802, 516407.302 16…
1   259.464         0.0  POLYGON ((519552.998 164295.600, 519508.096 16…
2   145.390         0.0  POLYGON ((518107.499 167303.399, 518114.301 16…
3   268.506         0.0  POLYGON ((520336.700 165105.498, 520332.198 16…
4   187.821         0.0  POLYGON ((521201.203 169275.505, 521204.303 16…
```

[214]:
```
map_df.plot()
plt.scatter(x=cdf['longitude'], y=cdf['latitude'])
plt.show()
```



[215]:
```
cdf['borough'] = cdf['borough'].str.title()
cdf['borough']
```

[215]:
```
0                      Croydon
1                      Croydon
2                       Sutton
3                   Hillingdon
4                     Havering
                 …
7539      Barking And Dagenham
7540                   Enfield
7541    Hammersmith And Fulham
7542                    Ealing
7543            Waltham Forest
Name: borough, Length: 7543, dtype: object
```

[216]:
```
bdf = cdf.groupby(['borough'], as_index = False)['special_service_type'].count()
bdf.head()
```

[216]:
```
                borough  special_service_type
0  Barking And Dagenham                   188
1                Barnet                   303
2                Bexley                   204
3                 Brent                   200
4             Brentwood                     1
```

[217]:

```
bdf.to_csv('animal_grouped.csv')
```

[218]:

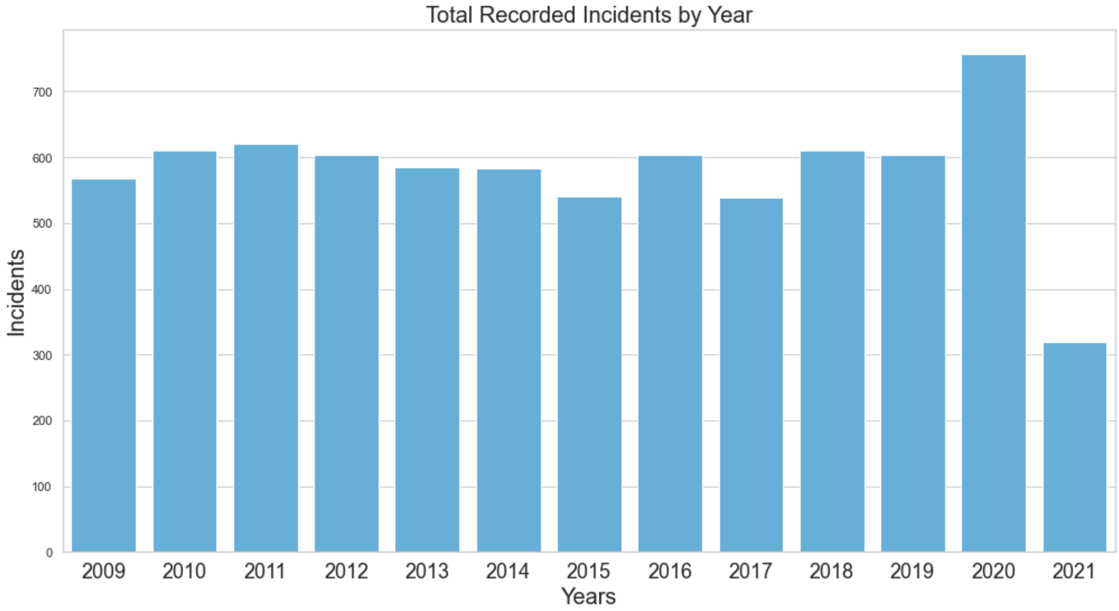
```
cal_gp = cdf.groupby(['cal_year'],as_index = False)['special_service_type'].
 ↪count()
cal_gp
```

[218]:

```
    cal_year  special_service_type
0       2009                   568
1       2010                   611
2       2011                   620
3       2012                   603
4       2013                   585
5       2014                   583
6       2015                   540
7       2016                   604
8       2017                   539
9       2018                   610
10      2019                   604
11      2020                   757
12      2021                   319
```

[219]:

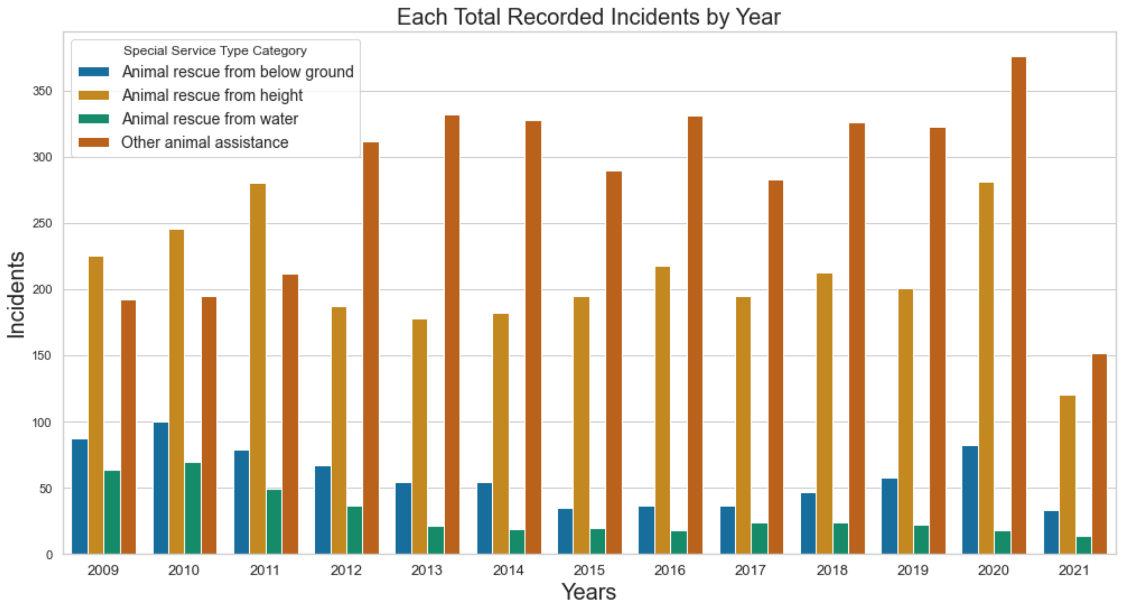
```
# setting grid color
sns.set(style="whitegrid")

plt.figure(figsize=(15,8))
# Plot the chart
ax = sns.barplot(x='cal_year', y='special_service_type', data=cal_gp,␣
 ↪color='#56B4E9')
_ = plt.xticks(fontsize=18, rotation=0)
ax.set_title('Total Recorded Incidents by Year', fontsize=20)
ax.set_xlabel('Years',fontsize=20);
ax.set_ylabel('Incidents',fontsize=20);

plt.show()
```



```
[220]: # group by top 3 service types per year
       calsstc_gp = cdf.groupby(['cal_year', 'special_service_type_category'],as_index␣
        ↪= False)['special_service_type'].count()
       calsstc_gp.head()
```

```
[220]:    cal_year    special_service_type_category  special_service_type
       0      2009  Animal rescue from below ground                    87
       1      2009         Animal rescue from height                   225
       2      2009          Animal rescue from water                    64
       3      2009           Other animal assistance                   192
       4      2010  Animal rescue from below ground                   100
```

```
[221]: # setting grid color
       pal = sns.color_palette('colorblind', 4)
       sns.set(style="whitegrid")
       plt.figure(figsize=(15,8))

       ax = sns.barplot(x = 'cal_year',y = 'special_service_type', hue =␣
        ↪'special_service_type_category', data = calsstc_gp, palette = pal)

       _ = plt.xticks(fontsize=13, rotation=0)
       ax.set_title('Each Total Recorded Incidents by Year', fontsize=20)
       ax.set_xlabel('Years',fontsize=20);
       ax.set_ylabel('Incidents',fontsize=20);
       plt.legend(loc='upper left', title='Special Service Type Category', fontsize=14)

       plt.show()
```



```
[222]: df_sstc = df.loc[df['special_service_type_category'] == 'Other animal␣
        ↪assistance']
       df_sstc.head()
```

```
[222]:    cal_year fin_year  pump_count  pump_hours_total  hourly_notional_cost  \
       0      2009  2008/09         1.0               2.0                   255
       1      2009  2008/09         1.0               1.0                   255
       4      2009  2008/09         1.0               1.0                   255
       5      2009  2008/09         1.0               1.0                   255
       6      2009  2008/09         1.0               1.0                   255

          incident_notional_cost final_description               animal_group_parent  \
       0                  510.0          Redacted                               Dog
       1                  255.0          Redacted                               Fox
       4                  255.0          Redacted                            Rabbit
       5                  255.0          Redacted  Unknown - Heavy Livestock Animal
       6                  255.0          Redacted                               Dog

                     property_type  property_category special_service_type_category  \
       0  House - single occupancy           Dwelling       Other animal assistance
       1                  Railings  Outdoor Structure       Other animal assistance
       4  House - single occupancy           Dwelling       Other animal assistance
       5  House - single occupancy           Dwelling       Other animal assistance
       6                      Park            Outdoor       Other animal assistance

                                   special_service_type borough_code  \
       0  Animal assistance involving livestock - Other …    E09000008
```

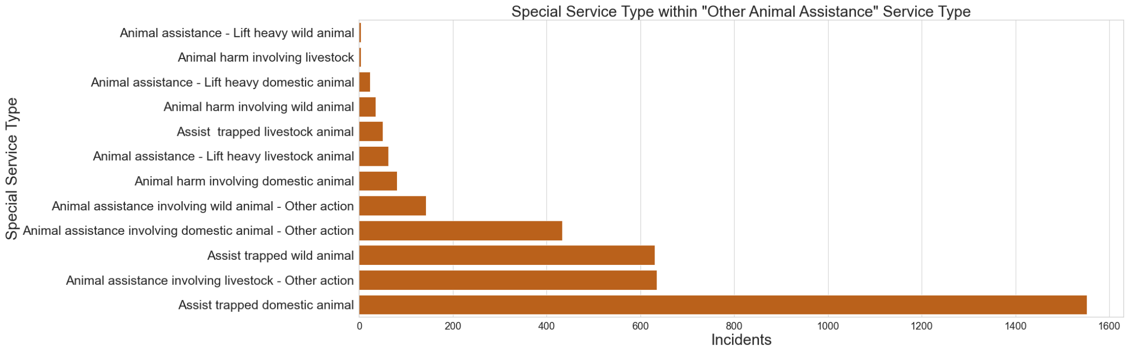
```
1  Animal assistance involving livestock - Other …    E09000008
4  Animal assistance involving livestock - Other …    E09000016
5  Animal assistance involving livestock - Other …    E09000002
6  Animal assistance involving livestock - Other …    E09000031

                borough   latitude  longitude
0               Croydon        NaN        NaN
1               Croydon  51.390954  -0.064167
4              Havering        NaN        NaN
5  Barking and Dagenham        NaN        NaN
6        Waltham Forest  51.557221   0.003880
```

[223]:

```
# where special_service_type_category = 'other animal assistance'
sstc_gb = df_sstc.groupby(['special_service_type'],as_index =␣
 ↪False)['special_service_type_category'].count()

sstc_gb = sstc_gb.sort_values(['special_service_type_category'])
```

[224]:

```
# setting grid color
sns.set(style="whitegrid")

plt.figure(figsize=(24,10))
# Plot the chart
ax = sns.barplot(y='special_service_type', x='special_service_type_category',␣
 ↪data=sstc_gb, color='#d55e00',orient = 'h')
_ = plt.xticks(fontsize=20, rotation=0)
__ = plt.yticks(fontsize=25, rotation=0)

ax.set_title('Special Service Type within "Other Animal Assistance" Service␣
 ↪Type', fontsize=30)
ax.set_xlabel('Incidents',fontsize=30);
ax.set_ylabel('Special Service Type',fontsize=30);

plt.show()
```



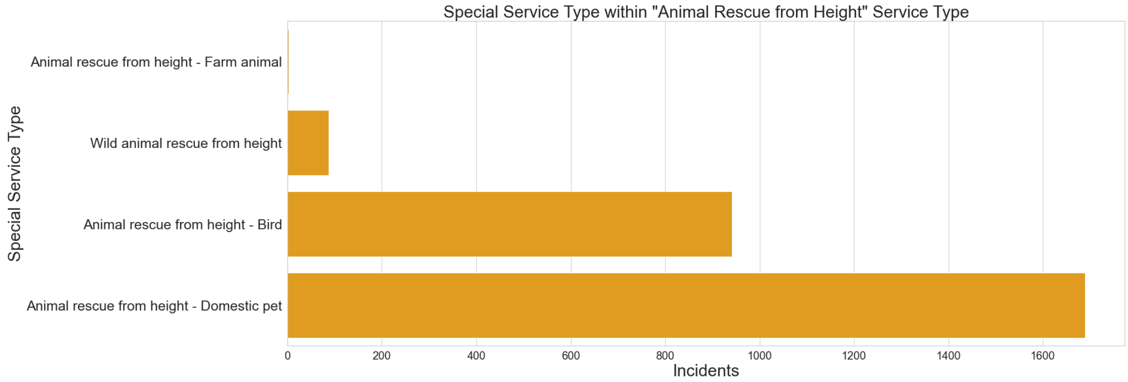
[225]:
```
df_sstc2 = df.loc[df['special_service_type_category'] == 'Animal rescue from␣
↪height']
df_sstc2.head()
```

[225]:
```
    cal_year fin_year  pump_count  pump_hours_total  hourly_notional_cost  \
8       2009  2008/09         1.0               1.0                   255
10      2009  2008/09         1.0               1.0                   255
14      2009  2008/09         1.0               1.0                   255
15      2009  2008/09         1.0               1.0                   255
16      2009  2008/09         1.0               1.0                   255

    incident_notional_cost final_description animal_group_parent  \
8                    255.0          Redacted            Squirrel
10                   255.0          Redacted                 Cat
14                   255.0          Redacted                 Cat
15                   255.0          Redacted                 Cat
16                   255.0          Redacted                 Cat

               property_type property_category special_service_type_category  \
8   House - single occupancy          Dwelling     Animal rescue from height
10                Tree scrub           Outdoor     Animal rescue from height
14                Tree scrub           Outdoor     Animal rescue from height
15                Tree scrub           Outdoor     Animal rescue from height
16                 Wasteland           Outdoor     Animal rescue from height

                     special_service_type borough_code  \
8             Wild animal rescue from height    E09000008
10  Animal rescue from height - Domestic pet    E09000006
14  Animal rescue from height - Domestic pet    E09000027
15  Animal rescue from height - Domestic pet    E09000008
16  Animal rescue from height - Domestic pet    E09000023

                 borough   latitude  longitude
8                Croydon        NaN        NaN
10               Bromley  51.403663   0.111722
14  Richmond upon Thames  51.457688  -0.295805
15               Croydon  51.392058  -0.062970
16              Lewisham  51.457382  -0.045121
```

[226]:
```
# where special_service_type_category = 'other animal assistance'
sstc_gb2 = df_sstc2.groupby(['special_service_type'],as_index =␣
↪False)['special_service_type_category'].count()

sstc_gb2 = sstc_gb2.sort_values(['special_service_type_category'])
sstc_gb2
```

```
[226]:                       special_service_type  special_service_type_category
       2   Animal rescue from height - Farm animal                              3
       3             Wild animal rescue from height                             87
       0           Animal rescue from height - Bird                            942
       1  Animal rescue from height - Domestic pet                           1689
```

```
[227]: # setting grid color
       sns.set(style="whitegrid")

       plt.figure(figsize=(24,10))
       # Plot the chart
       ax = sns.barplot(y='special_service_type', x='special_service_type_category',␣
        ↪data=sstc_gb2, color='orange',orient = 'h')
       _ = plt.xticks(fontsize=20, rotation=0)
       __ = plt.yticks(fontsize=25, rotation=0)


       ax.set_title('Special Service Type within "Animal Rescue from Height" Service␣
        ↪Type', fontsize=30)
       ax.set_xlabel('Incidents',fontsize=30);
       ax.set_ylabel('Special Service Type',fontsize=30);

       plt.show()
```



```
[228]: df_sstc2 = df.loc[df['special_service_type_category'] == 'Animal rescue from␣
        ↪height']
       df_sstc2.head()
```

```
[228]:     cal_year fin_year  pump_count  pump_hours_total  hourly_notional_cost  \
       8       2009  2008/09         1.0               1.0                   255
       10      2009  2008/09         1.0               1.0                   255
       14      2009  2008/09         1.0               1.0                   255
       15      2009  2008/09         1.0               1.0                   255
```

```
16      2009  2008/09         1.0               1.0                   255

    incident_notional_cost final_description animal_group_parent  \
8                    255.0          Redacted            Squirrel
10                   255.0          Redacted                 Cat
14                   255.0          Redacted                 Cat
15                   255.0          Redacted                 Cat
16                   255.0          Redacted                 Cat

               property_type property_category special_service_type_category  \
8   House - single occupancy          Dwelling     Animal rescue from height
10                Tree scrub           Outdoor     Animal rescue from height
14                Tree scrub           Outdoor     Animal rescue from height
15                Tree scrub           Outdoor     Animal rescue from height
16                 Wasteland           Outdoor     Animal rescue from height

                        special_service_type borough_code  \
8             Wild animal rescue from height    E09000008
10  Animal rescue from height - Domestic pet    E09000006
14  Animal rescue from height - Domestic pet    E09000027
15  Animal rescue from height - Domestic pet    E09000008
16  Animal rescue from height - Domestic pet    E09000023

                 borough   latitude  longitude
8                Croydon        NaN        NaN
10               Bromley  51.403663   0.111722
14  Richmond upon Thames  51.457688  -0.295805
15               Croydon  51.392058  -0.062970
16              Lewisham  51.457382  -0.045121
```

[229]:
```
df['special_service_type_category'] # Animal rescue from below ground # Animal␣
↪rescue from water
```

[229]:
```
0                 Other animal assistance
1                 Other animal assistance
2         Animal rescue from below ground
3                Animal rescue from water
4                 Other animal assistance
                         …
7539              Other animal assistance
7540              Other animal assistance
7541            Animal rescue from height
7542              Other animal assistance
7543            Animal rescue from height
Name: special_service_type_category, Length: 7544, dtype: object
```

[230]:

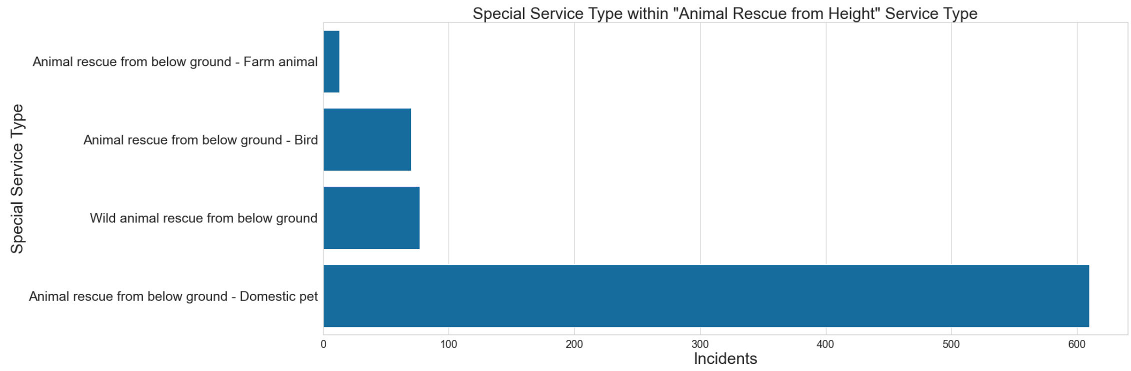
```
df_sstc3 = df.loc[df['special_service_type_category'] == 'Animal rescue from␣
 ↪below ground']
df_sstc3.head()
# where special_service_type_category = 'other animal assistance'
sstc_gb3 = df_sstc3.groupby(['special_service_type'],as_index =␣
 ↪False)['special_service_type_category'].count()

sstc_gb3 = sstc_gb3.sort_values(['special_service_type_category'])
sstc_gb3
```

[230]:

```
                               special_service_type  \
2   Animal rescue from below ground - Farm animal
0          Animal rescue from below ground - Bird
3            Wild animal rescue from below ground
1  Animal rescue from below ground - Domestic pet

   special_service_type_category
2                             13
0                             70
3                             77
1                            610
```

[231]:

```
# setting grid color
sns.set(style="whitegrid")

plt.figure(figsize=(24,10))
# Plot the chart
ax = sns.barplot(y='special_service_type', x='special_service_type_category',␣
 ↪data=sstc_gb3, color='#0072b2',orient = 'h')

_ = plt.xticks(fontsize=20, rotation=0)
__ = plt.yticks(fontsize=25, rotation=0)


ax.set_title('Special Service Type within "Animal Rescue from Height" Service␣
 ↪Type', fontsize=30)
ax.set_xlabel('Incidents',fontsize=30);
ax.set_ylabel('Special Service Type',fontsize=30);

plt.show()
```

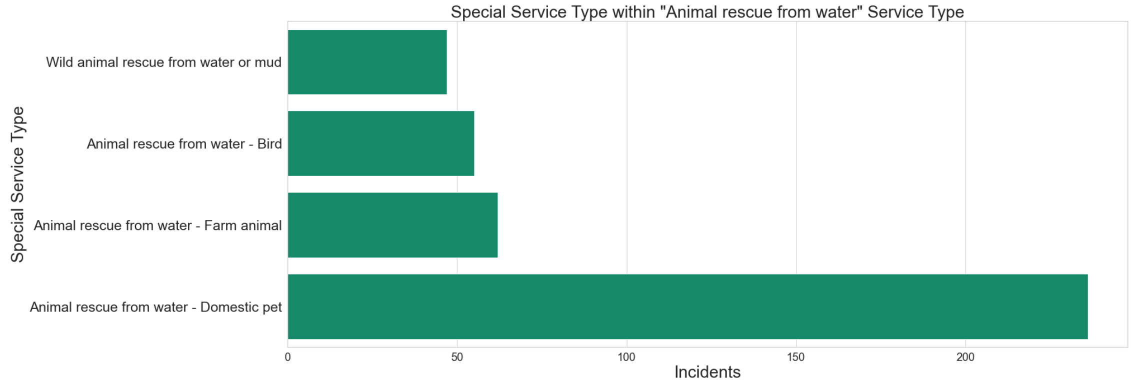


```
[232]: df_sstc4 = df.loc[df['special_service_type_category'] == 'Animal rescue from␣
        ↪water']
       df_sstc4.head()
       # where special_service_type_category = 'other animal assistance'
       sstc_gb4 = df_sstc4.groupby(['special_service_type'],as_index =␣
        ↪False)['special_service_type_category'].count()

       sstc_gb4 = sstc_gb4.sort_values(['special_service_type_category'])
       sstc_gb4
```

```
[232]:                   special_service_type  special_service_type_category
       3     Wild animal rescue from water or mud                             47
       0          Animal rescue from water - Bird                             55
       2   Animal rescue from water - Farm animal                             62
       1  Animal rescue from water - Domestic pet                            236
```

```
[233]: # setting grid color
       sns.set(style="whitegrid")

       plt.figure(figsize=(24,10))
       # Plot the chart
       ax = sns.barplot(y='special_service_type', x='special_service_type_category',␣
        ↪data=sstc_gb4, color='#009e73',orient = 'h')

       _ = plt.xticks(fontsize=20, rotation=0)
       __ = plt.yticks(fontsize=25, rotation=0)


       ax.set_title('Special Service Type within "Animal rescue from water" Service␣
        ↪Type', fontsize=30)
       ax.set_xlabel('Incidents',fontsize=30);
       ax.set_ylabel('Special Service Type',fontsize=30);
```

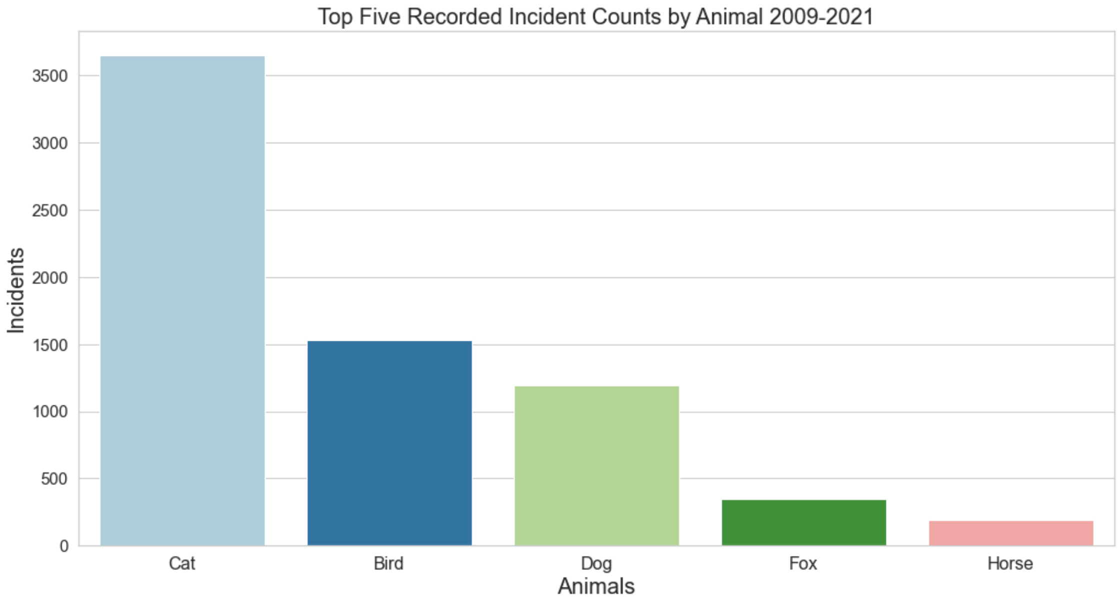
```
plt.show()
```



```
[277]: ssta = df.groupby(['animal_group_parent'], as_index = False,␣
        ↪)['special_service_type'].count()
       ssta = ssta.sort_values(['special_service_type'], ascending=False).head()

       ssta
```

```
[277]:    animal_group_parent  special_service_type
       3                  Cat                  3649
       0                 Bird                  1530
       6                  Dog                  1194
       9                  Fox                   349
       13               Horse                   193
```

```
[278]: # setting grid color
       sns.set(style="whitegrid")
       pll = sns.color_palette("Paired", 5)

       plt.figure(figsize=(15,8))
       # Plot the chart
       ax = sns.barplot(x='animal_group_parent', y='special_service_type', data=ssta,␣
        ↪palette = pll)
       _ = plt.xticks(fontsize=16, rotation=0)
       __ = plt.yticks(fontsize=15, rotation=0)
       ax.set_title('Top Five Recorded Incident Counts by Animal 2009-2021',␣
        ↪fontsize=20)
       ax.set_xlabel('Animals',fontsize=20);
       ax.set_ylabel('Incidents',fontsize=20);

       plt.show()
```



```
[258]: cat = df.loc[df['animal_group_parent'] == 'Cat']
       cat.head()
       # where special_service_type_category = 'other animal assistance'
       cat = cat.groupby(['special_service_type'],as_index =␣
        ↪False)['special_service_type_category'].count()

       cat = cat.sort_values(['special_service_type_category'], ascending = False).
        ↪head(1)
       cat
```

```
[258]:                            special_service_type  special_service_type_category
       8  Animal rescue from height - Domestic pet                           1484
```

```
[260]: bird = df.loc[df['animal_group_parent'] == 'Bird']
       bird.head()
       # where special_service_type_category = 'other animal assistance'
       bird = bird.groupby(['special_service_type'],as_index =␣
        ↪False)['special_service_type_category'].count()

       bird = bird.sort_values(['special_service_type_category'], ascending = False).
        ↪head(1)
       bird
```

```
[260]:                    special_service_type  special_service_type_category
       6  Animal rescue from height - Bird                            936
```

[263]:
```
dog = df.loc[df['animal_group_parent'] == 'Dog']
dog.head()
# where special_service_type_category = 'other animal assistance'
dog = dog.groupby(['special_service_type'],as_index =␣
 ↪False)['special_service_type_category'].count()

dog = dog.sort_values(['special_service_type_category'], ascending = False).
 ↪head(1)
dog
```

[263]:
```
              special_service_type  special_service_type_category
12  Assist trapped domestic animal                            409
```

[264]:
```
fox = df.loc[df['animal_group_parent'] == 'Fox']
fox.head()
# where special_service_type_category = 'other animal assistance'
fox = fox.groupby(['special_service_type'],as_index =␣
 ↪False)['special_service_type_category'].count()

fox = fox.sort_values(['special_service_type_category'], ascending = False).
 ↪head(1)
fox
```

[264]:
```
          special_service_type  special_service_type_category
10  Assist trapped wild animal                            159
```

[265]:
```
horse = df.loc[df['animal_group_parent'] == 'Horse']
horse.head()
# where special_service_type_category = 'other animal assistance'
horse = fox.groupby(['special_service_type'],as_index =␣
 ↪False)['special_service_type_category'].count()

horse = fox.sort_values(['special_service_type_category'], ascending = False).
 ↪head(1)
horse
```

[265]:
```
          special_service_type  special_service_type_category
10  Assist trapped wild animal                            159
```

[ ]:
```

```

[ ]:
```

```